OPEN SOURCE TIMES
开源时代
2009年9月刊 总第十二期

社区扫描：

自由软件基金会游说大企业 集体抵制Win 7
历届Linux内核会议中的数据

开源业界：

微软将Ubuntu列入Windows对手名单
欧洲开源发展领先世界

技术新知：

独辟蹊径J2EE网站之Tomcat篇
非LVM环境中的根调整

行业观察：

红帽CEO：开源与云计算是一对双胞胎
开源软件许可方式走向多样化

2009系统架构师大会盛大召开
——开放的力量

2009年8月28日至29日，北京歌华开元饭店，2009系统架构师大会（SACC2009）盛大召开。本次系统架构师大会由IT168、ChinaUnix、ITPUB、IXPUB共同主办。这场盛会汇集了开放系统架构及相关领域中的众多专家和学者，很多嘉宾都是来自知名企业的技术一线，有着丰富的行业经验和成功实践。

ChinaUnix.net

2009年9月刊 总第十二期

编辑出品：ChinaUnix.net

网络发行：ChinaUnix

主编：江晖
技术主编：樊强
执行编辑：周荣茂
内容编辑：周荣茂 覃里 唐川 周平
技术编委：高延斌 马路遥 白金
美术编辑：林在子
交流论坛：bbs.chinaunix.net
杂志下载：www.chinaunix.net www.itpub.net www.ixpub.net
联系我们：rmzhou@staff.chinaunix.net
投稿邮箱：rmzhou@staff.chinaunix.net

媒体支持：IT168.com

广告联系：温玉琴
电话：010-82658790
手机：13801339139
E-mail：wyq@it168.com

内容目录

行业观察

技术沙龙

技术新知

网友热评

致谢

本杂志得到ChinaUnix网站Linux时代社区版主的大力支持，技术文章大部分来自版主推荐，更多技术文章可以访问Linux时代精华区。
本刊分析评论部分文章来自IT168技术频道。

卷首语

****

很累，但很充实，2009 系统架构师大会已经在上个月底完美谢幕，在大会现场遇到了很多《开源时代》杂志的读者朋友，很是高兴，至少这一年来，杂志没有白做，至少还有这么多朋友在看，在关注。

先来说说这次系统架构师大会，本次大会是由 IT168 组织，在 CU、ITPUB 和 IXPUB 三个社区基础上召开的。这场盛会汇集了系统架构及相关领域中的众多专家和学者，很多嘉宾都是来至知名企业的技术一线，有着丰富的领域经验和成功实践，并且，最宝贵的是，很多特约嘉宾都带来了各自在多年从业经历中总结出的优秀实践材料，到这里与各位业内朋友们分享。

其实最初的时候，大会的主题定在 Linux 平台下的系统运维和架构，但是感觉到不够大气，经过网友和会务组的讨论，最终才确定为系统架构师大会，其实更多意义是指开放平台下的系统架构和实践。既然是基于开放架构，那么其基础就是开源软件，从本次大会的议题中我们也可以看出，比如来自新浪的开发平台架构介绍、来自 51.com 的 MySQL 数据库技术讨论等等，这些都和开源社区和开源软件密不可分的。

另外，我们还要提到的嘉宾是章文嵩先生，作为 LVS 项目的创始人，他也来到了大会的现场，做了题为《LVS 在行动》的主题演讲，博得现场听众的一致好评，会议结束后，他被热心的听众围堵在会场门口，就可见一斑。虽然现在 LVS 项目的代码变动很少，而且架构不再发生变化，但是它的应用范围却在不断的扩展，搭配其他的开源软件技术，在互联网应用需求爆炸式的增长环境下，不断取得性能上的新高度。

本次大会的数据库和存储技术讨论也是听众关注的热点，两场分论坛均座无虚席，这也和两场分论坛的主持人不无关系。在数据库架构设计分会场，由 Oracle ACE Direct 盖国强主持了该分论坛。来自业界的 4 位专家淘宝网技术总监、首席 DBA 陈吉平、Sybase 中国公司售前技术总监宋一平、新浪数据库平台主管 邵宗文和搜狐高级 MySQL DBA 叶金荣，分别从 4 种不同的数据库产品结合各自在多年从业经历中总结出的优秀实践为大家进行了详细的阐释。在 29 日下午举办的存储分论坛上，乐视网首席技术官杨永强先生与大家分享了分布式存储网络在视频网站中的应用；F5 公司北方区技术经理杨明飞先生介绍了 F5 公司文件虚拟化解决方案；资深系统架构师田逸为大家介绍了自己在 Moosefs 分布式文件系统方面的应用实践；最后，51.com 数据库主管徐景春也展示了 51.com 应用 MySQL 数据库进行在线灾备的实践分析案例。支付宝网站首席架构师冯大辉先 生为本次存储专场担任主持人。

本次大会的另外一个亮点是，我们在大会中也举办了一个教育行业的技术研讨沙龙，得到了北京邮电大学和华东师范大学的大力支持，他们给我们带来这教育行业开源技术的应用前沿技术及在应用中积累的经验，这在国内尚属首次。我们也期望这样的面向高校和教育科研行业的开源技术讨论越来越多，而不仅仅是昙花一现。

有关大会的更多内容，请访问大会官方网站和 ChinaUnix 网站的技术总结文章及相关下载。

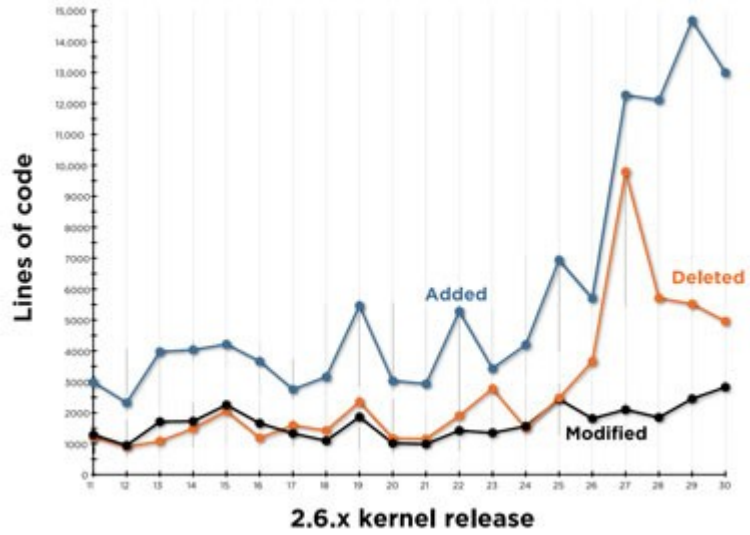
ChinaUnix 社区编辑：周荣茂

开源业界

****

报告称 Linux 开发人员在去年一年间增加了 10%

据国外媒体报道，Linux 基金会近日发布了名为“谁在编写 Linux”的年度报告。这份报告称，自 2008 年以来，Linux 的个人开发者人数增加了 10%。无疑，在全球经济衰退的背景下，开源软件无疑成为了许多企业的选择。



报告称，红帽（Red Hat）仍是对 Linux 内核贡献最大的企业。它负责 12.3%的内核内容。于此同时，占整个编写内容 6.9%的英特尔公司也正迎头赶上。此外该报告还提供了一些有趣的数据：

——自 2008 年以来，个人开发者的数目增长了 10%，这也反映了 Linux 在行业中的普遍程度；

——Linux 内核超过 70%的内容出自“受赞助的开发者”，即那些由红帽、IBM、Novell、英特尔、甲骨文等公司出资赞助的开发人员；

——自 2008 年 4 月以来，有 270 万行代码被添加进来。平均每天有超过 10923 行代码被添加进来。Linux 基金会称，“这种更新变革的速度，超过所有任何开源软件项目。”

——与增加代码相同重要的是删除代码。平均每天有 5547 行代码被删除，以保证内核代码的简洁与效率。

很明显，Linux 内核的编写方式选对了道路。充分考虑了能参与开发的人力，而未放弃质量上的种种优点。更重要的是，Linux 专家也从中获益。据统计，他们的收入也上涨了 50%。那么，微软对自家产品与 Linux 竞争感到焦躁不安，这样一种状况也不足为奇。想与 Linux 有效竞争，就相当于在同整个软件和硬件行业竞争。

Linux 或许不能代表计算的未来，但它确实证明了一群致力于此的爱好者究竟可以创造什么样的成就。借着超过 1000 名 Linux 内核开发者、200 家左右不同公司的携手努力，Linux 成为了开源社区力量的典范产品，同时也证明了如 Linux 基金会此类组织的价值。

雅虎资深工程师离职 加盟开源软件创业公司

据国外媒体今日报道，雅虎资深搜索及基础架构工程师道·卡廷（Doug Cutting）将于本月离开公司，加盟一家创业的开源软件管理公司 Cloudera。

雅虎与微软于 7 月 29 日宣布达成搜索合作协议。雅虎搜索业务目前有大约 400 名员工，根据合作条款，部分员工将前往微软，而其它员工将被辞退。因此雅虎最近出现了搜索和技术高管的离职潮。不过卡廷表示，自己离职与雅虎背离搜索业务没有关系。他说，“离职这件事酝酿已久，我绝不是出于抗议雅虎而离职。”

雅虎对卡廷在任职期间的工作表示感谢，在一份声明中表示，“我们祝愿卡廷及其新业务一帆风顺，期望雅虎能有机会投资 Hadoop，并与卡廷和不断壮大的 Hadoop 展开合作。”Hadoop 是卡廷开发的一套开源软件框架，可用于管理海量数据。雅虎和 Facebook 等多家大型网络公司均采用这个框架分析其收集到的大量信息。

Cloudera 今年初宣布，将把 Hadoop 引入企业界，以帮助各行各业的企业处理其不断增长的数据信息。Cloudera 高管认为，银行，石油公司、天然气公司和生化公司都将享受这套开源软件框架带来的好处。虽然本身是免费的开源软件，但 Cloudera 希望能借助 Hadoop 提供收费的咨询及支持服务。

卡廷将在 8 月底离开雅虎，并于 9 月前往 Cloudera 担任架构师。

IDC：开源软件市场提速

一份最新的 IDC 研究包括显示，在世界范围内开源软件收入每年将增长 22.4%，预计在 2013 年达到 81 亿美元。这次预测明显高于去年，IDC 列举了三个主要原因：

1、调查列表扩大，包括了更多的开源项目；

2、开源软件在过去的 12 个月，与以往表现出了更高的接受程度；

3、在 2008 年底，经济状况加速了开源软件的使用。

该报告还提到了以下几个重点：

1、大型软件巨头，如 IBM、SUN、Dell、HP、Oracle 等从他们对开源软件的积极支持中，获得了重要的收入。这将成为开源软件被接受和采用的主流。

2、混合商业模式看起来在增长，看起来很可能成为主流的商业模式。按需定制的厂商加入了 SaaS，SaaS 厂商提供按需定制服务，开源软件厂商销售衍生软件，同时闭源厂商提供更多的开源软件。

3、开源软件在各方面提高竞争优势，比如部分业务流程外包或者部分软件装置的需求正在增长，将帮助提高开源厂商销售的增长。

Mark Shuttleworth：桌面 Linux 难做也要做下去

马克·沙特尔沃思(Mark Shuttleworth)将手中的茶杯盖向上抛起，自己也跳了起来，窗外是蓝蓝的天空和朵朵白云。他在半空中伸手抓住了一起下落的茶杯盖，稳稳地着了地。此时，他的脸上露出了孩童般的笑容。当然，现在他可是在地球上；如果在失重的太空中，他完全可以做得更潇洒——7 年前，作为全球第二位也是非洲第一位自费的太空游客，他登上了俄罗斯“联盟”号宇宙飞船和国际空间站，在太空中度过了难忘的 10 天。

没有不可能

为了圆自己由来已久的太空梦，马克一共花了 2000 万美元，他还自掏腰包请了 20 多位亲朋好友前往俄罗斯的太空训练基地，让他们和自己一起体验宇航员的太空生活。

这可不是一笔小的开销，不过对于马克来说，这也算不上什么。实际上，他在翱翔太空之前的经历也颇具传奇色彩。父母是外科医生和幼儿园教师的马克出生于南非，就读于南非开普敦的一所普通大学，拿到了金融及信息系统的学位。还在自己 22 岁的时候，他就创立了一家名叫 Thawte 的数据安全公司。第二年，他拜访了当时最大的浏览器公司网景，成功地说服网景将自己的产品内置到网景的 Netscape 浏览器当中；随后，微软的 IE 浏览器也采纳了他的产品，这可能也是马克第一次与微软做生意。

“其实我更像是个搞研究的而不是做交易的。但是，当我发现互联网正在改变商业的时候，我马上就投身其中。”马克很快就获得了高额的回报。1999 年，在拒绝了一家公司高达 1 亿美元的报价之后，马克将 Thawte 以 5.75 亿美元的天价卖给了 VeriSign。那一年，他才只有 26 岁。随后，马克成立了自己的风险投资公司，开始在全球各地投资，他的个人资产目前也已经增值到了 10 亿美元以上。

一位还不到 30 岁的亿万富翁还能够做些什么？“卖出公司之后，我并不希望一种纸醉金迷的生活，我可以选择做一些在别人看来不可能做到的事情。”

他选择了遨游太空。2001 年，他从为数不多的既有钱又有闲、还有一副好身板的亿万富翁中被挑选出来，进入俄罗斯太空基地，进行遨游太空前的强化训练。“是的，我从小就对各种看起来不可能的事情感兴趣。”

2002 年 4 月，宇宙飞船成功升空，随后与国际空间站成功对接，马克终于有机会从太空中仔细端详自己生活的地球了。“是的，地球确实是圆的，是蓝色的，非常漂亮”。令他吃惊的是，即使是从太空的高度鸟瞰蓝色星球上的陆地，也很难再找到大片的自然美景了，除了大片的城市建筑和庄园，他甚至还看到了恼人的被污染的大气层。

“我非常幸运能够去太空体验，站在太空看一些事情，才发现生命的短暂、世界的渺小。当我回到地球以后，觉得更应该去做一些对人类、对世界有着正面影响的事情。”他花了两年时间思考自己未来的投资方向。

“我不是英雄，这也不是我的命运，但是我确实非常敬仰一些英雄人物，他们能够追逐自己的梦想，坚持自己想要做的事情并且坚定地做下去，而不管面对多大的困难和世俗的批评。”马克很佩服两个人：一个是达尔文，当他提出进化论而被广泛质疑的时候还能够坚持真理；另一个就是圣雄甘地，知道自己要做什么，一直坚持做下去。

2004 年，马克创立了致力于普及桌面 Linux 的 Ubuntu 社区。在南非古老的祖鲁语中，Ubuntu 的意思是“对他人的博爱”；与此同时，他还成立了一家商业公司 Canonical，专门负责支持

Ubuntu 项目的运营。第二年，他又投资 1000 万美元创立了 Ubuntu 基金会，从此将自己 90%的时间投向了 Ubuntu。为此，他经常乘坐自己的私人飞机在全球飞来飞去，鼓吹推广 Ubuntu，这架庞巴迪全球特快被命名为“Canonical One”，飞机的侧面涂上了一只被称为“Norman”的可爱的绿色小龙，它也是马克旗下的风险投资公司的吉祥物。

“你看，我已经过上了一种与众不同的生活了，对吧？我是亿万富翁、大学生、宇航员，生活不可能再好了。做 Linux 极客，对于我来说应该是个很好的平衡。”

与微软创始人比尔·盖茨的理想相似，马克希望每一台电脑里都能够装上 Ubuntu，这在几年前看起来几乎是不可能完成的任务。就在他创立 Ubuntu 的同时，另外几家主流的 Linux 厂商如 Red Hat 和 Novell 却宣布放弃了普及桌面 Linux 的努力，而将全部精力投入到能够带来更好收益的服务器领域。

“我对消费者不能亲眼看到的软件不感兴趣。”马克毅然决然地将 Ubuntu 的主攻方向放到了桌面系统上，他好像没有考虑过失败的后果，“既然 Ubuntu 能够把我的兴趣爱好和财务回报完美地结合在一起，为什么我不去做呢？”

尝试一切可能

但是，要在每台个人电脑中都装上 Ubuntu，谈何容易。从目前来看，微软在桌面操作系统的领导地位仍然是牢不可摧：Windows 仍然占据了全球桌面操作系统 90%以上的市场份额，苹果的 Mac OS X 分食了 5%，包括 Ubuntu 在内的 Linux 阵营虽然已经在桌面领域取得了长足的进步，也不过才突破了 1%的生死线，这也使得马克的事业更像是“乌托邦”。

“太空人”马克是梦想家，更是一名冷静的现实主义者。“很多人问我将来适合做什么？我很遗憾帮不上忙，我的回答是你应该每天去看看世界正在发生怎样的变化，自己去想想怎么去做，这个才是最重要的。”

互联网日新月异，新商业模式层出不穷，马克发现微软那种卖软件拷贝的商业模式已经摇摇欲坠。虽然目前桌面 Linux 还不成熟，但是它相信那一天终究会到来。目前，Linux 已经在服务器端占据了至少 10%以上的市场份额，为什么它就不能占领每个人的 PC 呢?

关键要解决的还是 Linux 的易用性的问题。脱胎于 Unix 的 Linux 继承了命令行的用户界面，非常适合程序员和极客们使用，却让普通消费者望而生畏。马克创立的 Ubuntu 就是为了更好地解决这个问题。与其他 Linux 版本不同，Ubuntu 有着绚丽易用、不亚于微软 Windows Vista 的用户界面。Ubuntu 的全部程序包只有 2G 的存储容量，可以通过自启动 U 盘安装。如果你想使用 Ubuntu，提出申请之后，你就可以得到邮寄的安装光盘，光盘里不仅仅只有一款操作系统，还包括办公软件、浏览器、email 等各种常用软件，而每半年(固定在每年的 4 月和 10 月)Ubuntu 还会发布更新的版本。更重要的是，所有这些都是免费获得的。

依靠这些特质，刚刚推出没几年的 Ubuntu 超越了其他 Linux 老大哥，成为桌面 Linux 当之无愧的领头羊。如今，Ubuntu 在全球已经有超过 1000 万用户，谷歌全球的 2 万名员工中有一半在使用 Ubuntu 的“变种”版本 Goobuntu，法国和西班牙等欧洲国家的政府和学校也开始大量采购 Ubuntu。早在两年前，惠普、戴尔、宏碁等主流 PC 厂商已经开始在笔记本电脑中预装 Ubuntu。

精明的马克还在团结一切可以团结的力量。Ubuntu 已经与英特尔主导的 Moblin 社区达成协议，将在热销的上网本中预装 Ubuntu 操作系统和 Moblin 用户界面和应用模式。马克觉得，Ubuntu 在上网本市场上有很大的机会。

就在不久前的 7 月 22 日，美国开源大联合(OSA)正式成立，主要成员包括

AMD、Canonical、Google、Novell、Oracle 和 Red Hat 等公司，身为南非人的马克被推举为 OSA 指导委员会委员和三位领导人之一。

作为支持 Ubuntu 的商业公司，Canonical 的盈利看起来还是遥遥无期——它将最高版本的 Ubuntu 软件免费提供给用户，然后希望这些公司能够将服务器和桌面的管理交给自己以收取服务费用。此外，Canonical 通过与戴尔等 PC 厂商联合开发定制的 Ubuntu 来获得收入。据马克透露，Canonical 的年度收入不超过 3000 万美元。

但是，马克并不着急；实际上，他正在谋划一场更大的棋局，那就是实现 Linux 阵营的大联合。他其实并不是 Linux 的生手。在创立 Ubuntu 之前很多年，他就已经活跃在另一个老资格的 Linux 版本 Debian 的社区里了，也正因为他在 Linux 社区里倾注的大量热情，他在软件开发者中享有很高的声望，被他们称为 SABDFL(自封的仁慈大君)。

因此，他很早就意识到，要想与强大的微软抗衡，Linux 阵营就必须尽快解决版本不一、四分五裂的问题——要知道，全球各种 Linux 版本有 300 种之多，这常常使得用户无所适从。

“希望 Ubuntu 的下一个长期支持版(LTS)能够与 Debian 同时发布。”马克表达了与老东家合作的决心，这次双方可能会在明年 4 月或者 10 月同时发布基于同一个 Linux 内核(Kernel)的不同的 Linux 新版本。与此同时，马克也在呼吁其他主流 Linux 厂商加入进来，大家基于同一个 Linux 内核、同一个图形界面标准开发 Linux 版本，这样就能够做到求同存异，共同发现问题，共同解决问题。

“我个人当然希望 Ubuntu 能够成为统一后的 Linux 标准。但是从实际情况考虑，我更愿意看到一家独大和百花齐放之间的平衡。因为一家独大后往往就会出现问题，从而阻碍创新。”马克认真地说道，“我可不希望 Ubuntu 成为另一个微软。”

在马克位于伦敦切尔西的豪宅当中，已经安装了传输速率高达千兆级的光纤。“我并不是用它来看高清色情电影的，我只是想看看光纤到户会是什么感觉，我这么做会有哪些不同。”他总在不断地尝试一切可能。

UNIX 服务器市场份额降 唯独亚太区增长

根据国外媒体报道，IDC 和 Gartner 两大分析公司对于第二季度服务器销售收入报告表明，全球范围内的销售数字反映了 Linux 和 Windows 平台销量的持续增长以及 UNIX 系统销量的再次下降。

来自于 IDC 的报告显示，Windows 服务器的销售在 2007 年第二季度上升了将近 18.7%，大约有 50 亿美元的资金被大型商业硬件厂商所垄断。根据发布的数据显示，服务器市场中有 38.2%份额是采用 Windows 平台的。Linux 系统占据了服务器总销售额的 13.6%——增长了 19%，这和 Windows 的增长幅度相仿。

与此同时，UNIX 的销售下降了 4%，现在的规模是 42 亿美元。这样看来，UNIX 服务器将大量的市场份额拱手让给了 Linux 服务器。

Gartner 的服务器市场研究分析师 Uko Tian 表示，UNIX 服务器市场的竞争如往常一样激烈，Sun 公司和 IBM 在收入增长方面保持着领先。“刀片服务器是这个季度中 x86 服务器平台主要的增长动力中非常特别的一个。惠普公司保持着第二季度内刀片服务器的领导地位，无论是在出货量还是收入方面。

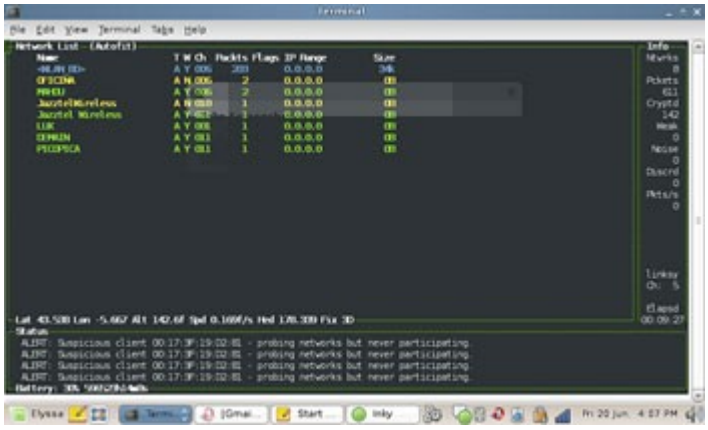
或许，亚太地区 UNIX 服务器获得增长的一个因素可能要归结为微软 OS 的垄断。几年来，微软不断远离政府和专业，并始终在全球都维持着一贯在授权和销售上的高压策略。大型硬件厂商，大

部分都已经学会了要行事低调——这是很多亚洲文化所推崇的。

同时，UNIX 很多优点是企业所青睐的，例如集中控制、更高的内在安全性和多平台能力等，这些都是客户需要的。此外，Sun 首席执行官 Jonathan Schwartz 推测到 2012 年末，这一地区将占到 Sun 公司总收入的 40%之多，这是目前这一地区交易总量的两倍还多(服务器的租用和托管）。

Linux 首次出现 Wi-Fi 漏洞 危及笔记本

一款主要的 Linux Wi-Fi 驱动程序中被发现存在一个缺陷，它使黑客能够控制笔记本电脑，即使笔记本电脑没有在 Wi-Fi 网络上。



据 Techworld.com 网站报道称，目前，Linux 平台上的 Wi-Fi 驱动程序还不太多，这显然是第一个可以被远程利用的 Wi-Fi 缺陷。据来自法国电信旗下 Orange 的研究人员劳伦特称，它影响被广泛使用的基于使用 Atheros Wi-Fi 芯片组的 MadWi-Fi Linux 内核设备驱动程序。劳伦特发现了该缺陷，并在上个月的 Black Hat 黑客会议上发布了相关信息。

劳伦特表示，如果不以手工方式修正 MadWi-Fi 驱动程序，用户就存在受到攻击的可能性。在披露该缺陷前，他向 MadWi-Fi 开发团队通报了相关资料。MadWi-Fi 开发团队已经发布了补丁软件。劳伦特说，但是，并非所有的 Linux 版本都集成了这一补丁软件。

据劳伦特称，该内核堆栈溢出缺陷使黑客能够运行恶意代码，即使笔记本电脑没有登录在 Wi-Fi 网络上，该缺陷也可能为黑客所利用。

Linux 用户此前一直受困于 Linux 驱动程序的短缺，要求在 Linux 内核中支持无线网络。由于 Wi-Fi 网络上的 Linux 笔记本电脑较少，安全专家和黑客一直没有足够的时间来关注 Linux 驱动程序，但是，在内核一级处理远程数据在 Linux 上与其它操作系统上一样容易造成问题。

微软将 Ubuntu 列入 Windows 对手名单

据国外媒体报道，微软近日在提交给美国证券交易委员会(SEC)的 10-K 年度管理文件中首次承认，诸如红帽(Red Hat)、Canonical 等 Linux 操作系统发行商已成为微软 Windows 客户端业务的竞争对手。

微软表示，红帽和 Canonical(注：该公司为 Ubuntu Linux 桌面操作系统发行商)大力进入上网本业务领域后，将同微软 Windows 客户端业务直接争抢市场，并有可能在常规笔记本业务领域对

Windows 构成潜在威胁。

美国独立市场研究公司 Directions on Microsoft 分析师罗布·赫尔姆(Rob Helm)表示：“在上网本领域，Linux 等操作系统已经撕开了微软 Windows 的防线，今后还有可能在常规笔记本和台式机领域向 Windows 发起挑战。”

在此之前，微软仅承认红帽将对微软服务器和商务工具业务构成威胁。微软在其 10-K 年度管理文件中写道：“目前微软 Windows 客户端业务正面临来自其他途径的激烈竞争，竞争对手包括苹果、Canonical 和红帽等。”

微软承认，在上网本业务领域，Linux 已受到了部分用户的欢迎，尤其在全球新兴市场，各大 PC 厂商出于压缩产品成本考虑，已计划在上网本中预装 Linux 操作系统。微软强调指出，作为微软合作伙伴的惠普和英特尔，也决定支持 Linux 杀入 PC 机市场。

美国科技博客作者托德·比肖普(Todd Bishop)首先注意到了微软 10-K 文件中对 Linux 市场威胁的描述。Directions on Microsoft 的赫尔姆对此表示，由于上网本业务利润较低，微软并不愿意大力进入该市场，而希望以“轻薄常规笔记本”来取代上网本。

赫尔姆还指出，自微软当前 Windows 客户端产品 Vista 上市以来，出现了大量硬件、软件不兼容现象，这也是导致 Linux“乘虚而入”的重要原因之一。尽管如此，目前在上网本市场中，仍是微软 Windows Xp 操作系统占据绝对优势地位。

Mac、Linux 的份额格局将产生变化

据国外媒体报道，日前，互联网数据统计公司 Net Applications 正式宣布，将改变操作系统和浏览器的市场份额的计算方式。据分析，计算方式改变之后，未来操作系统和浏览器市场份额格局都产生了变化，统计结果大大不利于苹果。

Net Applications 是一家核心的互联网情报公司，其市场份额的统计数据主要来自全球 1.6 亿个计算机网络。然而，近日，Net Applications 觉得该公司的统计方法太过“美国中心化”，继而决定改变统计计算方法。

Net Applications 在一份声明中表示：“在我们以后的报告中，我们将实行“国家一级加权制”，这就意味着我们将以流量比例为基础去调整我们的报告，我们将会以“这个国家中的总流量 VS 互联网用户数量”去统计互联网使用数据。”

Net Applications 指出：“例如，虽然以前我们有来自中国的重要数据，然而相对于中国的互联网用户数量来讲，我们的统计范围还是相对狭隘的。因此，在我们的全球统计报告中，我们将会着重关注中国的流量比例变化，这种变化将会使得我们的全球统计数据向着精确化的方向发展。”

据了解，在采用新的统计方法之后，苹果操作系统的市场份额自 9.77%下降至 4.73%，苹果的 Safari 浏览器也自 8.4%下降至 4.21%，而 iPhone 的市场份额的下降比率更是高达 50%左右。

相比之下，Linux 操作系统的市场份额有所提升，自 0.99%上升至 1.17%，而微软的市场份额的上升比率则比较平滑，没有出现太大的波动。

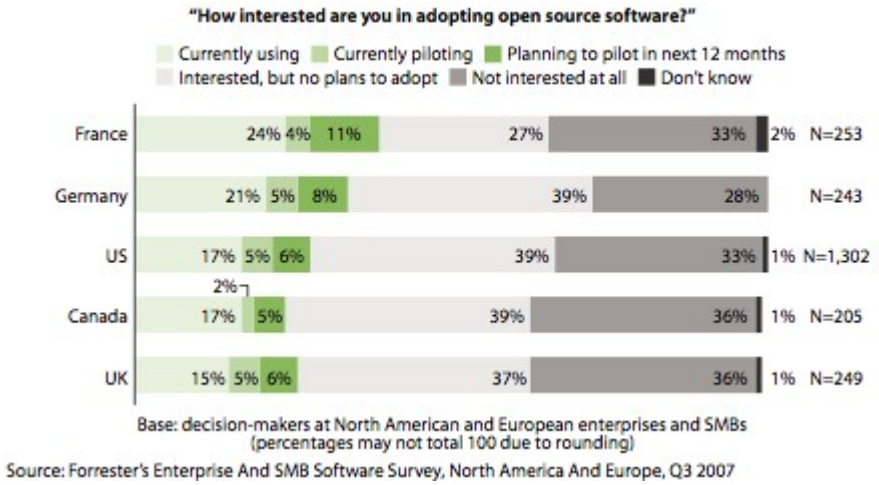
对于谷歌公司来讲，Net Applications 新变更的统计方法也是一个较坏消息。因为在 Net Applications 采用新统计方法之后，中国的百度搜索引擎的市场份额上升至 9%，这个上升幅度对谷歌搜索引擎的市场份额产生一定影响。

欧洲开源发展领先世界

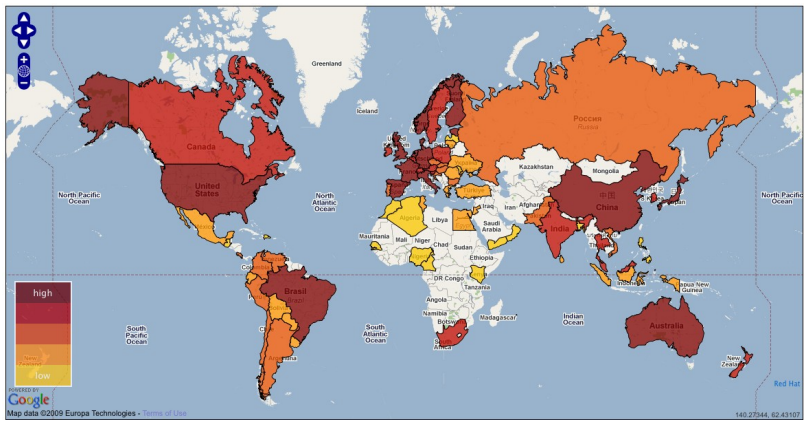
开源软件的采用和开发，欧洲领先世界。开源软件解决方案在欧洲有更大的市场，无论是服务器领域还是桌面领域。

世界最大的开源托管网站 sourceforge 上面的程序员只有 18%在美国，而有 33%在欧洲。欧洲在开源领域创造了 56 万 5 千个岗位，以及超过 2600 亿欧元的年收入。

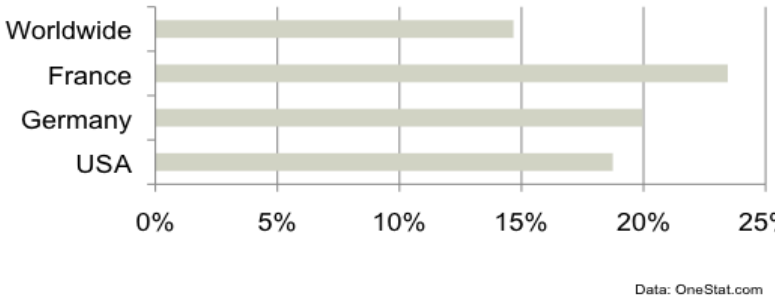
以下是相关对比分析图表：



Forrester 的研究报告---法国和德国领导开源的采用。



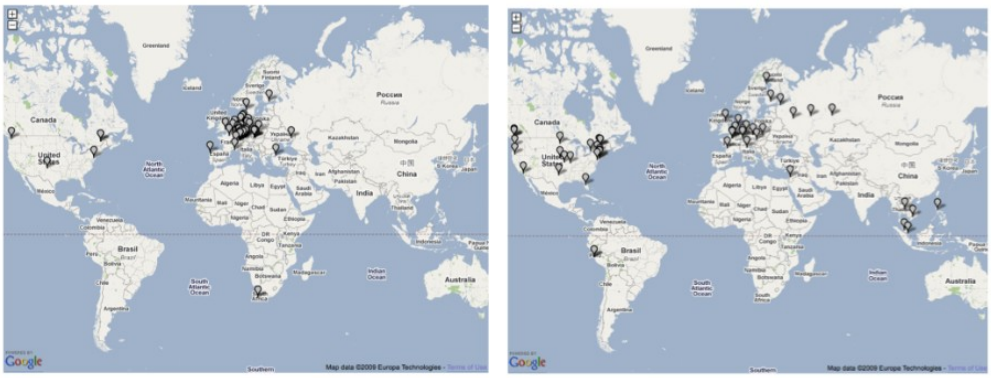
全球开源活跃地图（redhat）---欧洲遥遥领先，其次为美国、巴西、中国等。



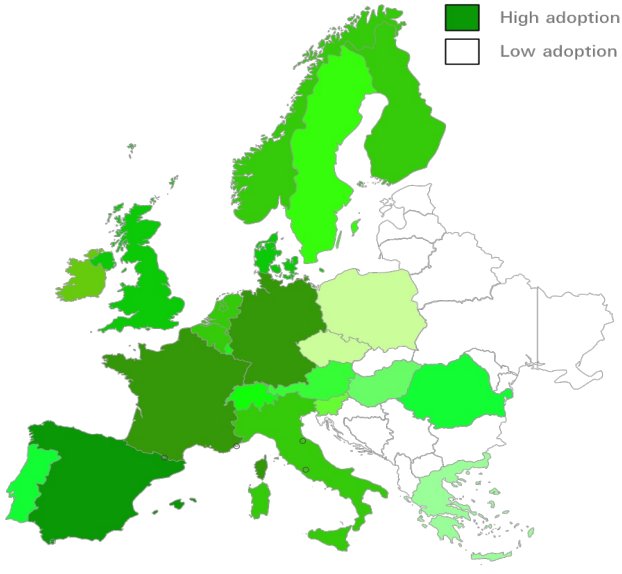
火狐市场份额对比图（08 年 11 月-09 年 3 月）

Credit: ohloh

Debian 开源贡献者分布图（基于 Ohloh 网站）



TYPO3 和 Drupal 用户分布图（基于 Ohloh）



欧洲开源软件采用分布图

Vmware 斥资 4.2 亿美元收购开源公司

据国外媒体报道，虚拟化设备提供商 VMware 在本月中旬宣布，将斥资 4.2 亿美元收购一家私营开源公司 SpringSource，增强云计算应用管理和开源社区等业务。

VMware 表示，收购 SpringSource 后，其业务将包括虚拟化、应用架构和云计算等。Vmware 在一份声明中称，这笔收购将花费 3.62 亿美元的现金和股票，及总计 5800 万美元的期权。交易预计将于今年第三季度完成。

在周一的分析师电话会议中，VMware 重申了对第三季度和本财年全年的业绩预期，以反映这笔交易。不过，预计这笔交易将影响 VMware 第三季度非美国会计准则运营性毛利率。

业内人士认为，收购 SpringSource 将有利于 VMware 的“平台即服务”计划。VMware 计划推出整合的“平台即服务”解决方案，其中包括 SpringSource 的软件和 VMware 原有的 vSphere 云操作系统。

SpringSource 是一家已成立 5 年的公司，拥有活跃的开源开发者社区，以及一些大公司客户。Vmware 表示将继续“坚持使 SpringSource 解决方案获得欢迎的原则，即 SpringSource 软件与多种中间件软件的互操作性，以及对开发者社区非常重要的开源模式”。

收购 SpringSource 使 VMware 站在了一些关键领域的前沿。SpringSource 的 Spring Framework 支持一半的企业 Java 项目，该公司同时为广泛使用的 Java 应用服务器 Apache Tomcat 提供超过 95%的漏洞修复工作。SpringSource 还拥有 Hyperic 应用监控工具。

IBM 和 Novell 釜底抽薪 SCO 资产禁售无力诉讼

近日，一位破产法院法官驳回了 SCO 集团提出的出售其部分业务的请求。出售部分业务可能有助于 SCO 继续与 Novell 和 IBM 打官司。

SCO 自从 2007 年以来一直由破产法院监管。SCO 提议要把该公司的大部分 Unix 业务出售给一家名为 Unxis 的公司。这是 SCO 提出的一系列建议中的一个最新的建议，旨在允许该公司退出破产保护，继续进行它的 Unix 法律诉讼。

另一方面，IBM 和 Novell 已经要求清算 SCO 的资产，从而有效地阻止针对他们的诉讼。

美国破产法院法官 Kevin Gross 本周三驳回了双方的建议。这位法官任命一个委托人控制 SCO 的资产并且评估 SCO 应该采取什么行动才能退出破产保护。这位委托人还将评估 SCO 赢得 Unix 诉讼案的机会。

Gross 法官说，这个法律诉讼的潜力必须要考虑到成本的现实。这个委托人没有 SCO 管理层的个人的和情感的投资，适合做出这个评估。

SCO 在 2003 年对 IBM 提起诉讼，指控 IBM 在 Linux 中的 Unix 代码侵犯了 SCO 的知识产权。

然而，在 2007 年，一位法官发现 Novell 拥有这些知识产权，而不是 SCO 拥有这些知识产权。由于这个裁决，Novell 对 SCO 提出起诉，要求 SCO 返还它向 Sun 和微软收取的 Unix 许可证费。

Gross 法官指出，SCO 自从 2007 年以来提出了许多重组计划，然后又撤销了这些计划。他说，这个最新的销售建议在是否有充分的业务理由方面存在着疑问，从而引起了各方的“善意”的怀疑。

SCO 建议以 525 万美元的价格出售其 Unix 业务，仅保留其移动应用程序业务。Gross 法官认为，这个建议是没有价值的。他说，SCO 的官员把该公司的赌注压在了法律诉讼上。如果允许出售 Linux 资产，法律诉讼就会成为 SCO 的“唯一业务”。

SCO 在声明中称，它正在评估这个裁决和自己的选择。

松下 NEC 推 9 款 Linux 手机 LiMo 系统将迅速普及

据无线 Linux 组织 LiMo 称，日本松下和 NEC 公司星期二推出了 9 款新的采用开源软件 LiMo 操作系统的手机。

自从谷歌和苹果在过去的两年里进入移动市场以来，手机市场的重点一直在向软件开发转变。手机厂商和运营商越来越多地寻求采用 LiMo 等开源软件替代产品以便降低成本。手机软件平台市场是诺基亚的 Symbian 操作系统占统治地位。但是，在过去的一年里，Symbian 的许多市场份额已经输给了苹果和黑莓厂商 RIM。

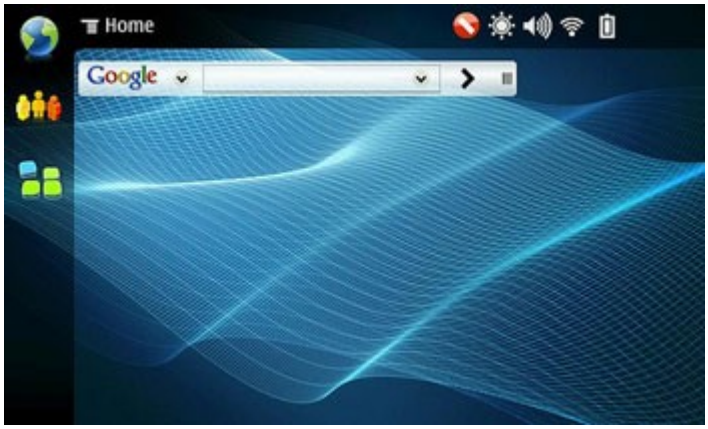
计算机操作系统 Linux 到目前为止还没有在手机市场获得成功。但是，Linux 的角色正在随着 LiMo 平台增长。而且谷歌正在其 Android 平台中使用 Linux。LiMo 称，日本移动运营商 KDDI 和触摸屏公司 Immersion 已经加入了这个非盈利组织。但是，LiMo 已经失去了一些最大的手机厂商的支持。到目前为止，只有 NEC、松下和摩托罗拉等小一点的手机厂商推出了一共 42 种采用 LiMo 操作系统的手机。同时，除了诺基亚之外，所有的主要手机厂商都表示将生产采用谷歌 Android 操作系统的手机。全球第二大和第三大手机厂商三星电子和 LG 电子也是 LiMo 的成员。但是，这两家公司到目前为止还没有推出采用 LiMo 操作系统的商品手机。

LiMo 希望利用它对电信运营商软件开发方面的更多的发言权来推广 LiMo 软件。LiMo 的关键成员沃达丰、法国电信旗下的 Orange、日本的 NTT DoCoMo、韩国的 SK 电信、Telefonica SA、美国运营商 Verizon 无线等公司已经保证在 2009 年推出 LiMo 手机。

传诺基亚欲放弃 Symbian 大力推广 Maemo 平台

据国外媒体报道，消息人士透露，全球第一大手机制造商诺基亚已决定放弃对 Symbian 手机操作系统的后续开发工作，转而大力推广其 Maemo 平台。

英国知名报纸《金融时报》德文版周二援引消息人士的最新表述称，虽然诺基亚已于 4 个月之前完成了对 Symbian 剩余 52%股份的收购工作，但鉴于 Symbian 平台性能已经老化，因此已经决定放弃该平台，转而推广该公司此前开发的 Maemo 平台。



Symbian 总部位于英国，诺基亚曾持有其 48%股份，其余股份由索爱、三星等主流手机制造商持有。诺基亚去年 6 月宣布，将出资 2.64 亿欧元(约合 4.1 亿美元)收购 Symbian 其余 52%股份，然后再专门为此建立一个开源基金会，使 Symbian 成为一款可免费使用的开源手机操作系统，从而同谷歌 Android 等平台争抢市场。

消息人士周二表示：“Symbian 平台已严重老化，而无法同其他性能更为先进的平台展开市场竞争。如此一来，诺基亚已决定放弃 Symbian，并改为大力推广 Maemo 平台。”诺基亚此前开发了 Maemo 平台，并计划将其主要用于诺基亚的上网设备当中。

对于诺基亚放弃 Symbian、改推 Maemo 平台的说法，诺基亚还没有作出正式回应。该公司一位发言人表示，不会对外界传闻加以评论。

明智之举

美国科技博客网站 TechCrunch 认为，如果诺基亚放弃 Symbian、改推 Maemo 平台的说法属实，应该说是明智之举。虽然诺基亚已为收购 Symbian 其余股权付出了 2.64 亿欧元，但好在 Symbian 的具体推广工作还没有实施开来，否则诺基亚今后将蒙受更多经济损失。TechCrunch 指出，在全球智能手机业务领域，苹果、RIM 和谷歌 Android 已严重威胁到诺基亚的市场地位。而要成功击退这些竞争对手发起的挑战，诺基亚就必须考虑推出一款性能更为先进的手机平台。

《金融时报》德文版援引消息人士的表述称，由于 Symbian 的编程代码已经落后，因此已严重影响到该手机操作系统的整体性能表现。Symbian 前身为 Psion 公司于上个世纪 90 年代开发的 Epoc OS 平台。另一方面，Symbian 共有 2000 万行代码，其代码数量快接近微软 Windows Xp 操作系统，因此已显得非常臃肿。

业界人士指出，正是因为 Symbian 代码老化、体积臃肿，导致该平台无法以更简单方式处理复杂手机任务。这也部分解释了为何在触摸屏技术使用上，诺基亚用了较长时间才解决该问题。另一方面，虽然诺基亚已解决了触摸屏技术，但其性能表现仍无法同 iPhone、Android 的类似功能相比。

值得注意的是，诺基亚同美国芯片巨头英特尔今年 5 月宣布，两家公司将联合开发一款名为

oFono 的手机操作系统。双方合作的主要内容是：oFono 将同时兼容诺基亚的 Maemo 平台和英特尔的 Moblin 平台，然后再向谷歌 Android 等平台发起挑战。

戴尔：Linux 机器遭退货并不是技术原因

在上月中旬举办的 OpenSource World 大会上，戴尔高级产品营销主管 Todd Finch 表示预装 Linux 的上网本的退货率和 Windows 上网本的退货率基本相同，这种现象并没有什么大不了的，是“不成问题的问题”。

微软首席运营官 Kevin Turner 在近日的微软年度金融分析师大会上表示，预装 Linux 的上网本的退货率极高，几乎是 Windows PC 机的 4 倍至 5 倍。显然戴尔并不赞同微软的看法，目前戴尔销售三款预装 Ubuntu 的笔记本，包括 Inspiron 15n、XPS M1330n 和 Mini 10v，根据自己的销售情况戴尔得出了上述结论。

Finch 表示，Linux 机器的退货并不是因为出现了什么技术问题，只是因为用户以为购买了一款 Windows 低价机型，使用时才发现是不熟悉的用户界面，“今后我们要在广告中说的更清楚才行”。

Finch 总结说：“在退换 Linux 机器的事件中我们并没有见到出于技术原因的，所以我们认为 Windows 和 Linux 退货率之间并没有显著差异，我们对 Linux 机器的稳定性和技术完备性感到十分满意。”

Linux 市场日益繁荣 免费应用会损害厂商收入

据市场研究公司 IDC 称，虽然经济衰退继续影响着首席信息官购买软件的胃口，但是，Linux 软件正在逆势增长。从 2008 年至 2013 年，Linux 软件销售收入的复合年增长率为 16.9%。到 2013 年，Linux 软件的销售收入将超过 12 亿美元。

正如 IDC 指出的那样，这种增长使 Linux 软件收入在 2013 年占全部软件销售收入的比例从 2008 年的 2.2%提高到 4%。这个比例是很小的。IDC 再一次考察了免费的 Linux 应用，并且披露了一些麻烦的数据。

人们一直以为 Novell 是 Red Hat 的主要的 Linux 竞争对手。根据 IDC 的数据，Novell 确实正

在日益成为 Red Hat 的真正威胁。但是，免费使用 Red Hat 的软件也许是更大的威胁。

Novell 在付费的 Linux 软件市场占 27.9%的市场份额并且在整个免费的 Linux 市场占 20.1%的市场份额。这个市场份额似乎不是很大。毕竟更多的人愿意使用免费的 Red Hat(包括 Fedora)，而不愿意使用付费的 Suse Linux 企业服务器软件。

然而，在增长方面，Suse Linux 增长速度更快。从 2007 年至 2008 年，Red Hat 的增长率是 1.9%，而 Suse Linux 的增长速度是 3.5%，几乎快了一倍。

从销售收入方面看，Novell 的市场份额是 29.8%，市场份额的增长率达到了 50.3%。而 Red Hat 的增长率是 14.8%。当然，Red Hat 的销售收入的基数比较大(3.195 亿美元)，而 Novell 的基数较小(1.126 亿美元)。但是，自从 2007 年以来，Novell 的 Linux 销售收入的增长速度一直超过了 Red Hat。

业内人士指出，企业使用开源软件计划的时间越长，他们就越不需要厂商提供的技术支持和维护。这样，免费的应用 Linux 最终会影响厂商的收入。

社区扫描

****

开源趣闻：历届 Linux 内核会议中的数据

Linux 内核会议是一个级别非常高的会议，每年举办一次。参与者主要来自 Linux 内核的官方开发者，开源社区黑客，以及与 Linux 相关行业的开发者，是一个纯粹的技术大会。

我们以 2001 年至 2008 年的合影留念照片为例，发现其中一些有趣的数据：首先这是一个男人的会议（历届中 98%以上都是男人），在女性参与方面，01 年 0 位，03、04、05、08 年各 1 位，06 年 2 位，02、07 年各 3 位（07 年其中一位是华人 MM）。再一下位长的比较的酷黑客（留长发，大胡子），01、03 年各 10 位，02 年 16 位，04 年 13 位，05、06、08 年各 9 位，2007 年 11 位。可见，有相当一部份的黑客喜欢留长发和大胡子(GNU 的创始人 Richard Stallman 胡子估计很难有人会超过他），最后统计一下光头（秃顶未统计在内）人数，只有 04、08 年各有 1 位参与。由于 Linux 内核会议是一个男人的会议，因此在这重点介绍介绍两位 Linux kernel 中的 MM。

Suparna Bhattacharya 基本上参加了历届的 Linux kernel 会议，她于 1993 年在 IBM 工作，她是 IBM 科学院成员，IEEE 高级会员。在 2000 年后，她参与 Linux kernel 工作，她是极少数女性 kernel 开发者之一。

Cao Mingming 在 07 年的合影照上可以看到，她在 IBM Linux 技术中心工作好多年。她对文件系统，存储，IO 和 kernel 可扩展性有着非常深的研究。

道德规范？GPL 软件只能免费？

两位 iPhone 应用程序开发者将多人跨平台 GPL 游戏 XPilot 移植到 iPhone，发布到 App store，收费 2.99 美元，同时按照 GPL 许可证要求在自己的网站上发布了游戏源代码。但 XPilot 的一位初始开发者认为他们收费的行为背叛了 GPL 的精神。

两人感到惟恐不安，认为 2.99 美元是合理的价格，收费主要是为了补偿他们的付出，以及支付 99 美元的开发者年费，他们所作所为完全遵循 GPL 的要求，并未破坏自由软件的道德规范。有评论认为很多 Linux 发行版和 OpenOffice 办公软件都以较低的价格出售，按许可证要求在网站上提供免费下载和源代码。

Linux、Twitter、Red Hat“赢”得Pwnie奖

第三届的 Pwnie Awards 已在赌城举行的 Black Hat 黑客大会上揭晓，获奖的详细名单已经公布，其中发现绿坝软件安全漏洞的密西根大学团队获 Pwnie for Mass 0wnage 提名，但最终未能获奖。

Most Epic Fail 授予了 Twitter/Google Apps，Twitter 在今年遭遇了跨站脚本攻击和 CSRF 蠕虫等问题，令人怀疑云计算的安全性；最佳歌曲奖授予了 Nice Report；终生成就奖授予了俄罗斯安全专家 Alexander Peslyak，aka Solar Designer，他公开了一系列攻击和安全保护技术，他的代码被 OpenBSD 和 Debian 在内的操作系统使用；Mass 0wnage 奖授予了 Red Hat，去年攻击者伪造了 OpenSSH 包签名.，入侵了 Red Hat 的服务器；Linux 开发团队授予了“Lamest Vendor Response”奖，因为他们将 Linux Kernel 的 Memory corruption 当成是拒绝服务（DoS）；微软操作系统上的 Conficker 蠕虫被授予了 Most Overhyped Bug，该蠕虫得到了媒体的广泛关注，但实际杀伤力微乎其微；Wei Yongjun 和 sgrakkyu 发现的 Linux Kernel FWD-TSN 块远程溢出漏洞，Sebastian Krahmer 发现的 Linux udev Netlink Message Privilege Escalation 漏洞，Ryan Smith 和 Alex Wheeler 发现的 ActiveX 漏洞等都得到了 Pwnie 的认可；最佳研究奖授予了 Bernhard Mueller，他不辞辛苦的演示了手机操作系统 Symbian 的一个安全漏洞。

雇用开源程序员的5个理由

近年来，开源已经渗透到 IT 的各个领域。今天经济环境的背景下，开源市场进入了加速阶段。虽然很多公司开始采用开源软件，但是开源程序员的数量并没有相应的增加。

为什么会是这样？一些公司担心雇用开源程序员将会是个不利-很可能使自己的专有软件面临开源风险。

让我们看看雇用开源程序员的 5 个不错的理由吧！

1.你可以看到比他们的简历更多的东西。因为他们从事的应用是开放的，你可以直接看到他们所写的代码。

2.开源程序员具有很好的警觉性，有意识及时为程序添加补丁。

3.虽然不能一概而论，但是开源程序员往往对其所从事的工作富有饱满的激情。这个看看那么多开

源程序员为开源社区做的伟大贡献，就能明白。

4.有了开源程序员，那么你会很高兴的看到他们的开源支持。不只是编程序，对终端客户的服务意识，早就建立了。

5.最后一个理由是，使用开源，你将省钱。既然利用开源，那自然应该雇用开源程序员。

雇用开源程序员有这么多的好处，IT 老板们还在等什么呢。开源程序员在 IT 招聘市场被认可和爆发，早晚的事儿！

UKGovOSS 上线：集合政府使用开源软件的情况

为了鼓励开展关于在政府内的开源和开源标准的讨论，Public Sector Forums 建立了一个新网站——UKGovOSS.org。这个站点的建立，其实是 2009 年四月在 PSF’s Local Government Open Source 会议上发起的讨论内容的延续。

为了运行这个站点，UKGovOSS.org 发表了一份关于当地政府开源使用情况的报告。这份名为“开源与否？”的报告，在采访了 168 位地方政府的管理和职员之后，数据显示有 370 个地方政府机关使用了开源。这份报告同时显示有 75%的理事会成员认为使用开源是因为费用低廉，大概有一半人认为在 2011 年会增加对开源的利用。UKGovOSS 认为，办公软件的开源，在未来的三年对地方政府影响最大。

Shuttleworth 向 Debian 捐赠 Canonical 雇员

最近 Debian 宣布了双年的发布周期，单数年的 12 月份开发冻结，后一年的上半年发布正式版。为了符合发布周期，下一版 GNU/Linux 6.0（代号 Squeeze）的技术冻结将提前到今年 12 月。这个时间表宣布之后，立即遭到了社区的激烈反对，Debian 随后推翻了原定的冻结计划，表示确切的时间表将在 9 月初宣布。鉴于 Debian 社区成员将此情况归咎于 Ubuntu，Mark Shuttleworth 决定采取措施，让一些 Canonical 公司雇员为 Debian 工作。

Debian 社区认为发布周期的决定是源于 Ubuntu vs.Debian 心态，认为该决定只会对两个发行版造成损害。Shuttleworth 在写给 Debian 邮件列表的电子邮件中解释了协调发布周期的原因，表示不仅关系到 Debian 和 Ubuntu，还关系到上游项目和其它发行版，它有利于 Linux 生态系统。为了在 12 月实现开发冻结，Shuttleworth 宣布让多位 Ubuntu 开发者为 Debian 项目工作。

开源项目越来越青睐 JavaScript

开源项目们都使用什么开发语言？一份由 Black Duck 发表的最新研究报告显示，开源项目越来越青睐 JavaScript，他们通过对项目中使用的各种开发语言的代码行数进行统计，发现 C 语言以 40%的比例遥遥领先，而 JavaScript 和 PHP 则呈明显上升趋势。

多数开源项目使用超过一种开发语言，Black Duck 的执行副总裁 Peter Vescuso 表示，他们发现，去年推出的开源项目中，36%的项目使用了 JavaScript。而从过去的 12 个月来看，JavaScript 代码的总行数占据了 7.6%的比重，在过去的 12 个月，上升了 2.1%。

使用代码行数比重进行统计是一种方式，这种统计方式对那些不够精炼的语言来说是有利的，Vescuso 说，为了公平，他们还使用了另外一种统计方式，某种语言在所有开源项目中使用的次数。在这种方式下，他们发现，过去的 12 个月中，所有开源项目中有 32%用到了 C 语言，而

JavaScript 则以 36%的比重领先，C++在过去的 12 个月见于 26%的开源项目，如果按代码量统计，C++的比重为 13%。

PHP 在过去的 12 个月的开源项目中的使用比例为 17%，按代码量统计其比重则为 5.2%，比去年上升了 0.3%。Black Duck 还做了一个统计，假如每个项目只统计一种语言（使用最多的那种），Perl 以 15%的比重领先，Java 和 C#以 11%的比重并列第二，JavaScript 则以 10%的比重居三。

对于非开源软件，Black Duck 的报告并未涉及，不过 Vescuso 表示，他们同很多开发者做过交流，他们表示，COBOL，C#以及 Visual Basic 等语言在非开源软件中被广泛使用。

荷兰黑客大会演示开源 GSM 网络

最近举行的荷兰黑客大会 HAR2009 上，德国自由软件黑客 Harald Welte 搭建并维护了一个开源 GSM 网络。

他在个人博客了公布了图片和技术细节，该开源 GSM 网络从荷兰监管当局获得运营执照，主要硬件部分是一对 BTS（无线电收发机基站），每个基站的发射功率是 100mW，天线安装在一棵树上，一台运行 OpenBSC 的 Linux 服务器提供基本的访问控制功能。系统的认证用户将收到一个 SMS，其中包含了国际移动用户识别码（IMSI）和认证标识符，总共有 391 位用户订阅了服务，他们的手机可以像在其它网络中一样使用。Welte 的成果证明 GSM 移动网络已成为黑客畅游的新领域。

开源企业远离 GPL

尽管 GPL 是开源软件使用最广泛的许可证，但开源企业却与这个对商业不友好的开源许可证渐行渐远。

开源跨平台程序开发供应商 Appcelerator 的 CEO Jeff Haynie 去年夏天作出一项决定，将该公司的产品从 GPLv3 换到 Apache。Haynie 并不是唯一一位远离 GPL 的商业导向开源社区成员。Black Duck Software 最近公布的一项报告显示，虽然 GPL 依旧是开源平台最具有统治性的许可证，但它正呈现下降趋势：开源项目采用 GPL 许可证的百分比从去年的 70%降至 65%。

Appcelerator 曾调查了大约 24 家开源软件开发商，发现只有一家仍然使用 GPL 系列许可证，其它公司或者是 MIT、Apache，或者是新 BSD。Eclipse 基金会的执行董事 Mike Milinkovich 表示，GPL 的拥护者喜欢告诉别人世界只需要一个开源许可证，他认为这是一种愚蠢的观点。Eclipse 基金会是目前众多提供更丰富商业条款开源许可证的机构之一。

开放是新的封闭

Android、Symbian、LiMo、Qt、WebKit.....等都是开源项目，但是它们到底有多开放? Visionmobile 的研究主管 Andreas Constantinou 解释了开源许可证和管理模式之间的差异，指出开源项目的管理模式在评估真正的开放性时被忽视误解了。

开源许可证如 GPL，LGPL，APL，EPL，MPL，BSD 和 MIT——规定了四大自由：获得源代码，修改、发行和捐赠代码。但是开源许可证只是故事的一半，另一半是源代码的控制管理。以移动行业为例，你可以将自己的代码加入到 Android 源码中，但最新代码的登记是否足够公开？谁决定 Symbian 源代码的变动？...这些问题常常没有正式答案，通常情况下开源项目的管理模式与开源提倡的精神是分道扬镳的，是封闭和私有的。比如开源促进会主席 Raymond 就曾批评自由软件基金

会(FSF)是大教堂（一般人的心中大教堂是指微软），因为它在控制什么代码进入 GNU C Compiler 或 GNU Lib C 上过于独断。开放是新的封闭。

开源大腕瑞科多将加盟 Facebook 任项目经理

据国外媒体报道，消息人士透露，开放源代码软件大腕、美国博客服务公司 Six Apart 高管大卫·瑞科多(David Recordon)已离职，并加盟 Facebook。瑞科多将出任 Facebook 高级开放项目经理。瑞科多在博客中表示，“过去一年我曾与 Facebook 团队密切合作，其产品、团队和创新给我留下了深刻印象。”

就在不久前，Facebook 还抱着自己的专有技术不放，当时招聘瑞科多这样的开放源代码大腕会显得有些滑稽。但 Facebook 正在发生变化，今年早些时候加入了 OpenID 基金会，使其相当大一部分开发者平台开放源代码化，Facebook Connect 服务获得了广泛认可。OpenID 基金会成立于 2007 年 6 月，其目的是开发和促进开放的互联网身份认证管理技术。Facebook Connect 服务；允许用户从其他网站访问 Facebook 数据。

瑞科多表示，在 Facebook 履新前，他不会披露有关这次跳槽的更多详细情况。瑞科多将于下周一在 Facebook 履新。Facebook 本月初收购了社交聚合网站 FriendFeed，主要目的就是获得后者的人才和技术。

自由软件基金会游说大企业集体抵制 Win7

据国外媒体报道，自由软件基金会(FSF)将举行一场抗议微软的活动，游说企业放弃 Windows 7，转而使用开源操作系统。

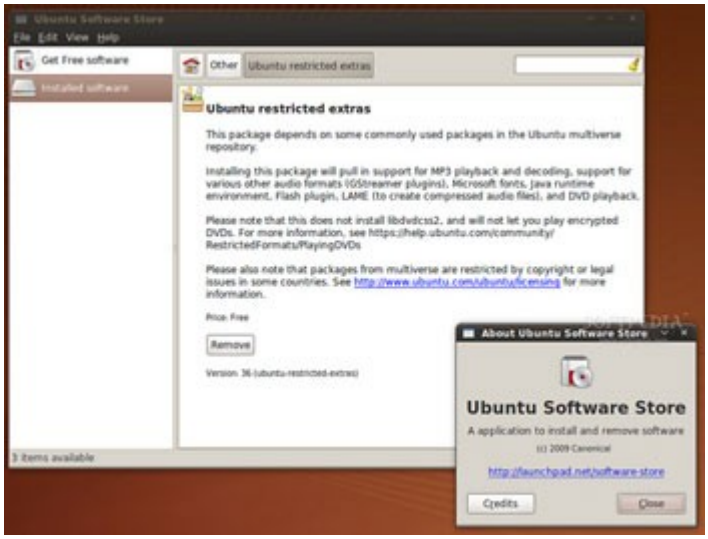
同时，自由软件基金会还将致信财富 500 强企业的 CEO，提醒他们，Windows 对企业的隐私、安全和自由是一个威胁。

此举正值 Windows 7 即将上市之际，但自由软件基金会执行董事彼得·布朗(Peter Brown)表示，此次活动不仅针对 Windows7，而是对微软整个商业行为的抵抗。

布朗还称，尽管此次活动主要针对微软，但自由软件基金同样担心其他一些产品，如本周五即将上市的苹果“雪豹”操作系统。

Canonical 透露 Ubuntu Software Store

Canonical 公开了新的 Ubuntu Software Store，通过软件商店，Canonical 希望能将所有不同的包管理器合并到单一的、统一的界面上。



这一雄心勃勃的目标不太可能在 Ubuntu 9.10 上实现。但 Canonical 希望它最终能取代更新管理器、Synaptic，janitor 程序及其它与包管理相关的程序，将所有这一切整合到 Ubuntu Software Store，让新的终端用户更容易上手，用单一的程序处理所有上述的功能。Canonical 计划 Software Store 1.0 版只提供基本的增加删除功能，未来的版本则慢慢取代 Synaptic、janitor、gdebi，甚至更新管理器。Software Store 使用 Python、GTK 和 Aptdaemon 编写，采用 GNU GPL v3 许可证，托管在 Launchpad 上。

行业观察

****

移动互联网：开源软件的又一个春天

随着中国 3G 时代的来临，移动互联网以其丰富多彩的应用成为广大企业挖掘的下一个金矿。作为新一代互联网，移动互联网主要是在开源软件和开放技术支撑下发展起来的，它的发展已经逐渐具备让任何人在任何地点、任何时间使用任何通信设备去做任何事情(5 个 Any)的能力。从移动互联网产业链上的广大企业来看，无论是从芯片制造商到终端设备商，还是从操作系统厂商到应用软件开发商，都已纷纷拥抱开源软件，将其列为自身发展的重要战略。

发展开源软件是大势所趋

近年来，开源软件在全球取得了长足的进步，基于开源软件的软件产品和服务日益成熟，应用也逐渐为用户所接受。专家预计，到 2012 年，全球 90%的企业和机构将有计划采用开源技术。

对此，Linux 基金会执行董事 JimZemlin 在接受采访时表示，从全球产业发展来看，Linux 面临着三大发展趋势：第一是 Linux 已经越来越多地被应用到各领域中，比如，许多手机操作系统是基于 Linux 平台开发的；在服务器市场，Linux 已经基本替代了 UNIX 操作系统；在高性能计算领域，Linux 已经占到 90%的市场份额。第二是当前全球经济下滑给 Linux 带来机遇，据某全球调查公司调查数据显示，全球有超过 75%的企业正在考虑或已经在使用 Linux 来降低成本。第三是产业融合的大趋势也给 Linux 带来发展机遇，比如现在笔记本电脑与智能手机二者之间在性能、移动性、便携性等方面都在逐渐走向融合，这给 Linux 带来巨大的发展空间。

中国开源软件推进联盟的统计显示，2008 年中国 Linux 销售额已经达到 3.2 亿元，比上年增长 16.6%。中国 Linux 操作系统的销售市场已从早期的政府教育领域开拓扩展到金融、邮政、电信、铁路、石油、航空等领域。

但我国开源软件产业也存在诸多有待解决的问题，比如在开源社区方面，许多社区还处于无序和无目标的状态，存在着人气不足、人才缺乏和项目缺少等问题。对此，工业和信息化部软件与集成电路促进中心副主任邱善勤认为，提高开源社区水平还需要解决以下几个问题：第一，资源整合问题，包括内容资源缺乏、开源软件类型单一以及技术人才分散等问题；第二，跨地域的技术开发问题；第三，统一技术标准问题，目前基于开源软件的数据库、中间件等商业产品很少，建立统一

的技术标准十分迫切；第四，开源软件商业化问题；第五，人才培育问题。

移动互联网成下一个金矿

由于移动互联网主要是在开源软件和开放技术支撑下发展起来的。从国际市场来看，众多的国际厂商纷纷加入到以移动互联网为背景的市场竞争中。如英特尔在发布了针对上网本市场的 Atom 处理器芯片之后，正在积极推动 Atom 向智能手机应用的拓展；智能手机芯片厂商 ARM 也在依托其芯片在功耗方面的优势向上网本市场寻求突破。

软件方面，谷歌在互联网上建立 Android 手机操作系统平台，目的在于掌握互联网，充分调控第三方在互联网上开发的免费的资源和应用程序。以英特尔公司为代表的企业联盟也推出了 Moblin 软件(Linux 操作系统)平台。诺基亚也已推出开放的互联网操作系统平台，微软也将在智能手机、上网本，以及相应操作系统云计算平台上推出其新成果。从中国市场来看，中国移动、中国联通、中国电信也在建立相应的手机平台，中科红旗、中标软件等均在开发上网本 Linux 操作系统。

对于移动互联网的开放性带来的机会，中国开源软件推进联盟主席陆首群日前在接受《中国电子报》记者采访时表示：第一，移动互联网的快速发展为开源软件的发展带来了良好机遇。第二，移动互联网的特点就是开放性，移动互联网的许多 IT 技术和通信协议主要是由开源技术构成的。第三，互联网上大量的知识积累和文化积累，极大地丰富了开源社区网站的信息内容和应用资源。但他同时指出，目前移动互联网还存在着安全认证等问题，这制约了网络交易的发展，而在移动应用的时候，相关通信协议也有待完善。

谷歌 Android 工程师苏哲也告诉《中国电子报》记者，现在的产业环境需要一个基于开放标准的开放系统平台，未来移动产业会建构在这个开放的平台上。因此，谷歌推出了基于 Linux 内核开发的 Andriod 操作系统平台，其中使用了很多开源组件，Andriod 在设计的时候就是以移动业务为中心，很多设计、理念都是面向移动设备。此外，谷歌还发起成立了开放手机联盟，一些移动设备制造商、软件开发商、中间件提供商以及电信运营商都参与其中，中国移动和中国联通也都是联盟的成员，联盟的目标是共同推动开放平台。

中国厂商需把握机遇

开源软件在给产业带来冲击的同时，也给中国 IT 厂商带来发展机遇。前不久，英特尔公司面向中国市场正式推出开放、开源的 Linux 操作系统平台 Moblinv2.0 公开测试版，该软件平台可使所有软件厂商基于它开发不同的应用、服务，其载体主要为移动互联网设备(MID)、上网本/上网机和嵌入式系统等终端。据悉，英特尔已经与中国本土的合作伙伴，包括中科红旗、中标软件、百资信息科技等公司，在 Moblin 平台上共同开发了一系列的操作系统，以及包括车载导航、视频、数字医疗、通信、游戏等各种应用。

在移动互联时代，软件世界里新铺了一条全新的跑道，大家还没有决出先后，美好的未来在等待大家共同开创。对于移动互联网给软件厂商带来的机遇，英特尔亚太研发有限公司总经理、中国产品开发总经理梁兆柱博士如是说。

在移动互联网以及开源的大趋势下，广大中国企业也积极参与其中，中标软件有限公司总经理韩乃平告诉记者，随着上网本的突飞猛进，移动终端已成为未来重要的发展趋势，中标普华操作系统在桌面技术基础上开发了面向上网本的定制系统，可以使人们更加方便地上网。此外，基于中标普华的操作系统，公司还开发了与高清机顶盒相关的定制系统，可以支持普通 CPU，以及高清和全屏技术，再加上浏览器、图形系统等，这其中充分利用了开源软件图形功能和浏览功能。

中科红旗软件技术有限公司总裁贾栋也表示，目前中科红旗的产品已广泛应用在桌面系统和服

务器领域，在嵌入式领域中重点突出的是 MID。目前红旗 Linux 在 MID 领域已经处于领先阶段，包括爱国者、联想和明基等厂商都已经选择了红旗的 Linux 作为他们 MID 产品的操作系统。贾栋还表示，在 MID 产品方面，公司将与消费类电子制造商和电信运营商深度合作。他说，这是公司发展的一个很重要的机遇。

削减 IT 成本将推动开源软件增长

根据 IT 市场研究公司 IDC 上周发布的预测显示，2008 年企业在在开放源码软件方面的支出增长了 34%，2009 年的开支可望至少再增加 24%。也就是说，将从 2009 年的 29 亿美元增加到 2009 年的 39 亿美元。这方面的支出主要包括基础软件、中间件及相关服务，但不包括嵌入式系统软件等设备。

IDC 公司的分析师 Michael Fauscette，该项研究的负责人说：“预计五年内，收入的平均的复合年增长率为 24.4%，在 2013 年达到 80 亿美元，相关的广告收入排除在外。”

Fauscette 在接受采访时说，加速增长的原因是企业寻求降低成本的解决方案。“经济不景气反而推动了开放源码的发展。”他补充说，如果今年实际的增长率超过了百分之二十四，他也不会感到惊讶。

虽然订阅 Linux 和系统相关的软件在最近几年推动了开放源码的支出，但 IDC 的最新的调查结果表明，开支增加其实是全方位的，包括数据库，应用软件和中间件。

“我们看到应用程序方面，特别是在中间件方面有着越来越多的绝对增长。”Fauscette 说。

一个典型的例子是 Ingres 公司，一家开源数据库软件系统供应商。其收入在 2007 年增加了一倍。2008 年，他们又增加了 32%，达到 6800 万美元。该公司说，客户在开放源代码解决方案方面的兴趣大大增加。

“我们已经看到了巨大的增长，现在，IT 企业压缩预算，使得我们的经济模式看到了巨大的利益。”Ingres 的总裁罗杰布克哈特在接受采访时表示。

开放源码软件的大量增长也正在推动软件作为服务(SaaS)和云计算基础设施的增长，根据 IDC 的 Fauscette 介绍。“云计算基础设施和服务供应商正积极推动数据库和中间件，因为这是真正能为他们保持规模并降低成本的关键因素。”他说。

分析人士指出，对于关键的云基础设施，特别是亚马逊、Rackspace 公司和谷歌以及较小的初创企业，依赖于开放源码软件。毫不奇怪，关键开源提供商已经把目标标准了云计算。

“开源软件继续铺平道路，例如，新的云供应商能够在线为现有的在职人员提供各种层次的云服务和差异化服务。”红帽公司首席技术官 Brian Stevens 在该公司上周的云计算在线论坛上说。

“开放源码软件和云计算的发展潜力巨大，二者相互补充。”451 集团分析师 Matthew Aslett 在接受采访时表示。

Aslett 也在红帽的论坛上表示，开源软件正呼吁那些想要将他们的系统移动到另一个云，再回到内部或混合两个系统。“这在方便企业转变他们的部署方面有一定的潜力。”他说。

红帽 CEO：开源与云计算是一对双胞胎

全球性的金融风暴已经波及到中国的大多数企业，这场风暴让企业面对的市场竞争日益激烈，也对其 IT 系统提出了更高的要求。而与此同时，企业又不得不面对日益压缩的 IT 开支。作为 IT 部门的负责人，CIO 如何才能在更低投入的情况下满足业务部门提出的更为苛刻的要求呢?

2009 年 8 月 5 日，红帽“开源.云计算”大会在北京举行，红帽首席执行官卫赫士(Jim Whitehurst)以及红帽技术专家对全球开源及云计算技术的最新发展趋势；用户如何用红帽的技术搭建公共云及私有云；以及红帽云计算相关的产品和技术等问题逐一详解。

卫赫士表示，开源是一种已经流行了多年的软件模式，其先进性已经接受了众多企业的检验，而云计算技术是当今最为热门的 IT 技术，代表了 IT 技术的未来发展方向。以 Google、wiki、facebook 为代表的全球 90%以上的云计算在开源之上运行的。开源与云计算是双胞胎，需要创新精神、参与的力量。

卫赫士认为云计算对传统软件商业模式构成了挑战：传统软件成本高，不断升级不断投入，而在云计算、开源基础上，成本更低，更加灵活。

“目前来看，私有云更有发展潜力，但公有云是长期的发展趋势，是更理性的选择。”卫赫士对硅谷动力记者说，“云计算迫切要解决一些技术问题，比如安全性、兼容性、稳定性等，还要解决协作和创新等问题。”

据悉，红帽为促进开源技术在中国进一步应用，发起了一项开源协作创新能力计划(OSCI)项目，日前，这一项目又取得了突破性进展：由广东省政府牵头建立的广东云计算中心选择红帽作为其技术提供商。

评论：从国外云计算的政府支持说开去

作为一种 IT 交付模式，云计算也许仍是新兴事物。但正如互联网诞生后被大学校园、美国政府大力推广一样，目前美国政府机构正在大力推行采用云服务或自行构建云服务的计划。他们的技术决策者似乎普遍持有这种观点：云计算的优点远大于风险；如果精心规划、认真实施，风险是可以缓解的。介于国内目前 50%的数据中心增长和改造投资来源于政府（另外 30%来自金融业），因而本文主要从美国政府所采取的云计算推动策略出发，谈谈国内云计算环境的建设。

优势远大于风险 国外政府积极推进云计算

和国内还停留在学术层面以及试商用层面的云计算环境不一样。国外的云计算理论已经较为成熟，网络上既有成熟的商用云计算，方案商也有成熟的企业/内部云架构的搭建方案。政府机构在其技术顾问的建议下正在积极的使用基于云计算的系统架构。美国哥伦比亚特区的前任 CIO Vivek Kundra 早在以前就使用过 Google Apps 的云计算服务，并给予很高的评价。目前，在他的推动下，美国国防信息系统部门（DISA）正在其数据中心内部搭建云环境，而美国宇航局（NASA）下设的艾姆斯研究中心最近也推出了一个名为"星云"（Nebula）的云计算环境。

但是，由于目前云计算的定义繁多，安全隐患等问题广惹争议，因此政美国国家标准和技术研究所（NIST）草拟了云计算的定义，以免在政府内实施云部署的人员偏离方向（后文会提到该定义）。而美国总务管理局（GSA）已经向云服务和云平台提供商们发布了关于当前云计算规模的信息质询（RFI），此举是为了在需求之前先摸清市场。

5 月份在华盛顿举行的联邦 IT 成本节约论坛（Federal IT On A Budget Forum）上，来自美国军方、国防信息系统局、总务管理局、宇航局、国家标准和技术研究所以及国防部、能源部和内政部的人士上台发言，充分讨论了所在组织采用或考虑采用云平台和云服务时所面临的安全、法规、互操作性和 IT 技能发展等问题。巧合的是，国内的第一个官方性质的云计算大会也是在 5 月 22 日召开，与会的几位院士和工信部官员分别对国内云计算发展的前景和问题做了阐述。可以看出，对于云计算从国内舆论到政府相关部门都给予了密切的关注，前阵子中化集团在 IBM 的助力下也成功部署全球第一套企业云计算系统。

制定云标准集中力量突破云安全难题

美国国防信息系统局（DISA）计算服务部的技术官 Henry Sienkiewicz 也表示出了乐观态度，他认为国防信息系统局能够克服云计算带来的种种不可避免的挑战。据了解，DISA 主要为国防部的军事部门提供 IT 服务，在 Sienkiewicz 的推动下，DISA 制订了一系列为实施基础架构即服务、平台即服务和软件即服务的战略。DISA 构建一个名为"快速访问计算环境"（Rapid Access Computing Environment）的内部云，这个云将让用户可以从自助服务式门户访问数据中心的资源，以下拉菜单方式列出了众多虚拟化的 IT 资源。它希望获得的诸多优点包括：降低 IT 成本、按使用量计费、加快部署大型机级系统、数据中心实现标准化以及可以灵活地向上及向下扩展。

而在众人关注的云安全方面，Sienkiewicz 提到了最近在用户访问和控制方面取得的"突破"。比方说，DISA 现采用这一种模式："租用服务资源的用户"必须遵守标准化的主机托管环境规则，因而继承了主机的访问控制。至于互操作性问题，Sienkiewicz 承认自己对如何建立这个互操作环境不太有把握，但是他认为只有提供一个标准的 API 接口才有可能解决问题。。他说："我们不知道谁有权来定义互操作性的标准，但我们不会让厂商来制定和控制这一标准。"

从以往经验来看，让一种新模式得到政府的接受需要经历繁琐的过程。然而一旦大家开始讨论云计算，在其定义的具体制定方面势必会引发激烈的争论--为了避免企业的恶性竞争，政府也就必须拿出明确的云计算定义。据美国国家标准和技术研究所（NIST）对云计算所下的定义（现已到了第 14 个草案版本），云计算至少拥有五个"基本特点"、三种交付模式以及四种部署模式。NIST 信息技术实验室的两名起草者 Peter Mell 和 Tim Grance 写道："这些定义、属性和特点将不断完善及变化。"

据业内人士透露，NIST 所给出的定义（共有 677 个单词）比业界一些缺乏系统性的定义来得全面，阐述了平台即服务与基础架构即服务的区别，还描述了有着共同利益的组织所共享的"社区云"（community clouds）。Peter Mell 说："我们是科学家，我们对包罗万象、而概念模糊的定义不满意。因此我们对云定义采取了分类的方法，这在定义中并非总是很常见。"

"一流厂商制定标准，二流厂商设计产品，三流厂商生产复制"。在云计算目前面临着概念模糊、隐患众多以及缺乏典型部署的情况下，我们也可以看到不少先机。率先梳理出云计算概念和标准，不仅仅可以推动信息化建设，更加有力的是在初期就对目前鱼龙混杂的云计算市场环境进行一次过滤。而从政府的角度出发，科学的发展观也正要求我们用前瞻性的目光把握战略性的变革。在 IT 架构中尽早的引入云计算，解决其带来的各类问题，无疑对于制定云计算的标准和概念有着很多好处。

先摸底，再实验——法规不是障碍

上个月，美国总务管理局（DSA）向基础架构即服务提供商们发布了信息请求，这又一次表明了政府机构对云越来越有兴趣。信息请求包括 45 个问题，涉及价格、服务级别协议、操作程序、数据管理、安全和互操作性等方面。

这类摸底主要是为了帮助政府机构更好的了解云计算的市场环境，从而制定相应的云标准。据 DISA 的 Peter Mell 表示，美国政府不会明确规定采用哪一种云标准，但确实认为自己是开发云标准的助推剂。他说："我们认为，数据和应用程序在云与云之间的移植性至关重要；我们还认为，拥有标准的云接口（API）以便从采用基于标准的机制的云来提供资源至关重要。"DISA 部门就致力于为移植性和互操作性找到最低标准。

DISA 还在密切关注云计算如何符合法规要求，最重要的是符合《联邦信息安全管理法案》的要求。Mell 表示，DISA 出版物中规定的安全控制虽然缺乏案例，但从内容上看并没有触犯法规。Mell 在联邦 IT 成本节约论坛上发言时称，尽管面临挑战，确保云安全还是"一件能做的事"。

事实上，很多专家都曾对云计算的合法性产生过质疑。原因在于共享的资源导致了隐私和拥有权遭到了破坏—没有一个公平的监管机制，你如何知道自己租用的资源（动态的）是否可以即时得到应用？而这也正是云计算需要制定标准的迫切需求之一。

与之相比，我国面对的云计算建设还处在雏形阶段，各种应用、厂商的介入以及学术界的碰撞越来越多的展开。而与之相对的法规还有不少空白，这就给了国内云计算环境一个有利的机会，在与美国情况截然不同的先天优势下（不用回避已有法规），自由的施展拳脚。而只有从实践的角度出发，摸清市场、做足尝试，再谈标准的制定，才能真正的切合产业需求—将云计算推动起来。

云计算是增加政府透明度的有力工具

除了在政策上重视云计算的有关标准制定以外，2010 年美国联邦预算的发布文件中着重概述了云计算计划--包括资助众多试点项目，这又一次表明了云计算在政府机构的 IT 政策和战略中会扮演越来越重要的角色。前任 CIO Vivek Kundra 在接受采访时说："2010 年预算中的云计算投资体现了政府渴望降低成本、推动整个政府机构进行创新的迫切需求。另一方面也希望我们可以为政府雇员提供一个在异地仍然可以使用的完全相同的办公环境。"

据介绍，美国政府最终可能希望把政府内部云连接到由共享数据、应用程序和 IT 资源组成的超级云。DISA 提出了美国政府的云基础架构：各个政府机构在其中拥有各自的云实例或节点。有了这种方法，就能开发出跨云节点运行的应用程序、创建更安全的环境、防御安全漏洞和攻击，并且使用专门为政府云设计的工具来集中管理云资源。不过，这一切工作也将依赖于云计算标准方面的进展。

政府行为一般都能带动产业界的跟进。亚马逊公司最近在华盛顿特区为与政府机构合作的 IT 服务公司开设了为期两天的有关云计算方面的培训课程。而与政府机构合作的微软联邦事务部门首席技术官 Susie Adams 表示，美国政府目前对微软的 Azure 云服务很感兴趣，他们认为这种方法可以迅速扩展 IT 资源，满足奥巴马总统下令加强透明度的要求。

与之对照，国内目前在加速推进政府信息化办公和办公透明度的过程中，如能参考云计算模式进行相应的改造，将为今后政府办公的便利性以及 IT 资源扩容打下坚实基础。不过，由于云计算在标准化、安全性和法律保障方面很容易遇到新问题，因此改造云环境的政府、企业的 IT 负责人就需要成为多面手。他们必须妥善处理安全和治理方面尚未得到解决的问题，又要规划好前方的道路。

削减 IT 成本将推动开源软件增长

根据 IT 市场研究公司 IDC 上周发布的预测显示，2008 年企业在在开放源码软件方面的支出增长了 34%，2009 年的开支可望至少再增加 24%。也就是说，将从 2009 年的 29 亿美元增加到 2009 年的 39 亿美元。这方面的支出主要包括基础软件、中间件及相关服务，但不包括嵌入式系统软件等设备。

IDC 公司的分析师 Michael Fauscette，该项研究的负责人说：“预计五年内，收入的平均的复合年增长率为 24.4%，在 2013 年达到 80 亿美元，相关的广告收入排除在外。”Fauscette 在接受采访时说，加速增长的原因是企业寻求降低成本的解决方案。“经济不景气反而推动了开放源码的发展。”他补充说，如果今年实际的增长率超过了百分之二十四，他也不会感到惊讶。

虽然订阅 Linux 和系统相关的软件在最近几年推动了开放源码的支出，但 IDC 的最新的调查结果表明，开支增加其实是全方位的，包括数据库，应用软件和中间件。“我们看到应用程序方面，特别是在中间件方面有着越来越多的绝对增长。”Fauscette 说。

一个典型的例子是 Ingres 公司，一家开源数据库软件系统供应商。其收入在 2007 年增加了一倍。2008 年，他们又增加了 32%，达到 6800 万美元。该公司说，客户在开放源代码解决方案方面的兴趣大大增加。“我们已经看到了巨大的增长，现在，IT 企业压缩预算，使得我们的经济模式看到了巨大的利益。”Ingres 的总裁罗杰布克哈特在接受采访时表示。开放源码软件的大量增长也正在推动软件作为服务(SaaS)和云计算基础设施的增长，根据 IDC 的 Fauscette 介绍。“云计算基础设施和服务供应商正积极推动数据库和中间件，因为这是真正能为他们保持规模并降低成本对关键因素。”他说。

分析人士指出，对于关键的云基础设施，特别是亚马逊、Rackspace 公司和谷歌以及较小的初创企业，依赖于开放源码软件。毫不奇怪，关键开源提供商已经把目标标准了云计算。“开源软件继续铺平道路，例如，新的云供应商能够在线为现有的在职人员提供各种层次的云服务和差异化服务。”红帽公司首席技术官 Brian Stevens 在该公司上周的云计算在线论坛上说。“开放源码软件和云计算的发展潜力巨大，二者相互补充。”451 集团分析师 Matthew Aslett 在接受采访时表示。

Aslett 也在红帽的论坛上表示，开源软件正呼吁那些想要将他们的系统移动到另一个云，再回到内部或混合两个系统。“这在方便企业转变他们的部署方面有一定的潜力。”他说。

Citrix：Novell 虚拟化联姻的唯一选择

商业 Linux 分销商 Novell 将在服务器和桌面虚拟化方面有怎样的举动？这个问题问的很好，这也是 Novell 高层至今还没有解决的一个问题。

Novell 在 2006 年 7 月发布了 SUSE Linux 10，成为第一家出货面向 Linux 的 Xen hypervisor 的商业 Linux 厂商。考虑到管理工具 Xen 的情况以及 Nobell 在嵌入 Xen 产品中提供除了 SLES 10 以外其他操作系统支持，所以很难说 Novell 这么做是否有些操之过急。Red Hat 当然会这么想。Red

Hat 将其 Enterprise Linux 5.0 嵌入 Xen 产品的发布推出到下一年 3 月，之后为了进一步考虑决定放弃 Xen 而选择去年九月收购 Qumranet 获得的 KVM hypervisor。基于 KVM 的 hypervisor，Red Hat Enterprise Virtualization 两个月前开始 beta 测试，预计将在今年年底面市。

那么 Novell 会怎么做？Novell 支持在 SUSE Linux 11 中支持 Xen，SLES 11 中包括一项 KVM hypervisor 的技术预览。但是没有人希望使用一个 Linux 发行版本来托管 Linux 和 Windows 图像。如果真是那样的话，Red Hat 可能就会被 Xen 羁绊住，独立的 hypervisor 解决了这一问题。而 Novell 并不具备。

VMware 具备，而且成效很显著——过去几年 ESX hypervisor 业务带来大约 10 亿美元的业务。VMware 在 4 月发布了最新的 ESX server 4.0 hypervisor 和相关的 vSphere 4.0 管理工具并从一个月之后开始出货，这些产品甚至比在 x64 服务器市场占据主导地位的上一代产品更加出色。

由此看来，ESX server 并没有受到 Hyper-V R2 和相关 Windows Server 2008 R2 操作系统的威胁。从技术方面来看，Hyper-V 并不会给 ESX server 带来太大威胁，但是事实上微软免费提供 Hyper-V 的策略的确给 VMware 带来不小的压力，因为你可以想象到所有人会向 VMware 讲述 Hyper-V 有多棒。这的确会让 VMware 总裁兼首席执行官、微软前高管 Paul Maritz 抓狂。

Citrix 在今年 2 月发布了免费的 XenServer 5.5 hypervisor，并从 7 月开始出货。这款产品同样也给 VMware 带来了价格上的压力。XenServer 5.5 有来自合作伙伴的工具帮助管理基于 Xen 的级虚拟机，这也是 XenServer 5.5 能够于 VMware 竞争的最大卖点。Citrix 斥资 5 亿美元收购了 XenSource，而 XenSource 正是 Xen hypervisor 项目背后的商业实体。虽然 Citrix 在上周宣布已经累积有 150000 明用户下载了免费的 XenServer，但是并没有表现这对收入有怎样的影响。

Oracle 也推出了他们的服务器虚拟化产品，收购 Virtual Iron 和交付了一系列包括 RHEL 的软件栈（称为 Oracle Enterprise Linux）。到 Oracle 完成对 Sun 的收购的时候，肯定会有 Solaris 和其他虚拟化技术加入市场竞争。

看上去 Novell 和 Citrix 是彼此需要的，就好像 Citrix 需要与微软联盟推出管理 XenServer 和 Hyper-V hypervisor 的 Essentials 管理工具，就像 Novell 需要与微软签订协议分销价值 3.4 亿美元的 SUSE Linux 支持合同。我猜测用不了多久我们就会听到 Novell 和 Citrix 宣布面向 SUSE Linux 的 Citrix Essentials、Novell 将 XenServer 5.5 hypervisor 作为单机产品的消息。

Novell 理论上也可以选择 KVM，但这只会有助于竞争对手 Red Hat，这还意味着 Novell 必须想办法开发他们自己的 SUSE 企业虚拟化产品。当然，Novell 可能会拉上 Oralce，改变成 Red Hat 虚拟化 Logo。但是这两种做法都不太体面。然后，就会诞生面向 SUSE 和 XenServer 的 Citrix Essentials。

开源效应：志愿者帮 Mozilla 完成了 40%的工作

很少有哪些机构利用开源模式的效果会超过 Mozilla，看了最近的关于 Mozilla 的报道：与其内部员工相比（下面将会强调），火狐浏览器及其他产品多少是由志愿者完成的，结果显示的数据非常惊人。

虽然 Mozilla 内部员工已经由 2005 年的 15 人增加到了 250 人，但是仍然有 40%的工作是由志愿者完成的，工作内容包括编写程序代码、设计火狐图标等。

相信大家都会问这些志愿者花费这么多时间，用金钱来衡量的话会有多少呢?

做道数学题

如果说 Mozilla40%的工作是由志愿者完成的，而 Mozilla 内部的员工有 250 人，这 250 人完成 60%的工作。我们可列这样的方程式 0.60x=250(x 为所有的有偿和无偿劳力)，那么就可以算出 Mozilla 有效的劳力（把志愿者也考虑在内）约等于 417 位全职雇员。这意味着志愿者相当于完成了 167 位全职员工的工作(417 中的 40%)。

每年 Mozilla 就可获利数百万美元。当然这是我们通过计算数字得出的，不过估计下这些所有免费的劳力的价值还是相当有趣的一件事。

开源热潮

毫无疑问 Mozilla 利用开源模式掀起了一股潮流，使得开源开发与闭源开发更加不同，人们不仅仅是能参加，而重要的是他们想参加。让人人都能参与到开发中当然很重要，因此 Mozilla 早已经实现人人都容易参与其中，包括编写程序和测试，甚至还开设了 Mozilla 软件营销社区。

有兴趣的朋友还可以看看其他类似开源公司中志愿者数字，不过 Mozilla 的志愿者比率不是一般的高。

开源软件许可方式走向多样化

去年夏天，Jeff Haynie 走到十字路口，作为开源跨平台应用开发软件供应商 Appcelerator 公司的 CEO，他做出了一件事关公司未来的重大决定。这个决定就是：不再使用开源软件领域知名度最高、最流行的 GPL（Gnu General Public License）软件许可，而是选择了更有利于公司开展经营活动、实现赢利的另一种许可方式。Haynie：“我们的产品最初采用的 GPL v3 许可证，但是去年夏天，我们决定换成 Apache。”

在开源社区，像 Haynie 这样为了公司经营而由 GPL 换成其他许可方式、或者正在考虑换成其他许可的人并不少见。今年 6 月，开源开发工具供应商黑鸭软件（Black Duck Software）曾进行过一次调查，它们发现，尽管迄今为止，开源软件基金组织（Free Software Foundation）发布的 GPL 尽管依然是最被人熟知、也是使用最多的软件授权许可方式，但是，其统治地位已经在开始衰弱。调查还发现，尽管采用 GPL v3 的项目数量上升得很快，但是采用由 GPL 衍生的其他种类软件许可的开源项目与前一年相比，从 70%下降到 65%。

Haynie 介绍说，在 Appcelerator 公司准备换掉 GPL 之前，他曾经调查了与他处于同一个市场的 20 多家软件供应商，他发现，只有一家采用的是与 GPL 有关的软件许可，“而其他公司使用的则要么是 MIT、Apache，要么是新 BSD。”他说。

“GPL 的支持者通常会告诉人们，这个世界只需要一种开源许可证，那就是 GPL，而我认为事实不是这样的，坦白地说，这是睁眼说瞎话。”Eclipse 基金（Eclipse Foundation）的执行董事 Mike Milinkovich 说。Eclipse 基金是众多提供开源许可证的组织之一，与 GPL 相比，这些组织发布的许可证更有利于商业经营者。在采用这些许可方式的企业看来，它们在源代码的发布条款方面更宽松（这意味着公司更容易挣钱）而且条款更清楚，它们的开发者社区成员更投入、素质更高。

GPL 妨碍了开发者挣钱

随着开源市场逐渐偏离原来的轨道，GPL 也就越来越不受欢迎。早期的开源软件开发者大多数是出于个人爱好，他们愿意把自己的劳动奉献出来，开发出有用的产品与人分享，而 GPL 的宗旨就是要鼓励这样的人、要鼓励人们的这种奉献精神。而今，根植在 GPL 深处的这种理想主义显得有些不合时宜，因为企业正在成为开源社区的主力，这些企业把开源开发作为一种赢利手段，而不是仅仅为了奉献。在这些企业看来，GPL 中关于代码修改方面的条款规定得过于苛刻，影响到企业经营，因此，不再愿意采用 GPL。

具体而言，企业拒绝 GPL 的主要原因是，其条款对企业通过修改源代码后编译成新的软件产品推向市场的行为进行了严格的规定。Eclipse 的 Milinkovich 认为，Eclipse 在这方面要宽松很多，是

GPL 最有力竞争者之一。他说：“我们制定许可方式的一个出发点就是有利于商业化。Eclipse 生态系统的一个非常典型运营模式是，利用 Eclipse 社区的现有技术，企业基于此加入一些自己的商业性元素，最后在 Eclipse 许可证之下把产品推向市场。”

一直在关注开源问题的法律界专业人士 Van Lindberg 认为，GPL 限制了企业通过对源代码进行修改然后从中赚钱的这种经营行为。“从本质上说，GPL 的原则就是，如果源代码来自 GPL 许可下的项目，此后对这些代码所进行的任何修改也必须适用于 GPL，也就是必须公开。”他说，“比如，你可以销售采用 GPL 许可的代码，但是前提是，你必须保证那些获得你代码的人有权利也有能力无需支付任何费用就可以把代码交给（或者出售给）其他人。”

Appcelerator 公司的 Haynie 解释说，他之所以从 GPL 换成 Apache 是经过几个星期的认真研究和思考的。“我们做出这一决定的主要原因是基于商业上的考虑。”他说，Apache 中取消了 GPL 关于公开源代码的规定，而且规定中也没有什么重大缺陷。

“我们认为 Apache 更有利于我们的经营活动，换句话说，更有助于我们挣钱。”他说，“而且，从法律的角度来说，其许可中各种专利条款写得非常明确，也让我们放心不少，而以前我们常常担心会有法律的纠纷，因为 GPL 的条款比较复杂难懂。”

“制定 GPL 的人都是理想主义者，他们特别强调自由软件的精神，也就是所有软件必须是自由的，自由地获得，自由地使用，为此甚至强迫让一些软件变成自由软件。”Milinkovich 说。

“有些人几乎把自由软件当成一种宗教信仰，它们认为开源软件只应该有一种许可，那就是 GPL，要么就不是开源软件。”GPL 的另一个竞争者 Apache 软件基金组织（Apache Software Foundation）主席 Jagielski 说。

研究机构 Info-Tech 公司的分析师 Howard Kiewe 表示，GPL 被认为是一种保证再发布者有完全版权的有效方法，“然而，与众多主要面向企业经营活动的许可方式比较起来，GPL 的这些规定已经有些不合时宜了。”

GPL 竞争者的优势

Jagielski 说，除了条款上更有利企业经营、实现赢利外，与 GPL 相比，其他的大多数许可更通俗易懂，表述得也准确。“很多人们在采用 GPL 时的确有过担心，因为它的表述比较难理解。”他说，“而你必须明白什么时候违背了 GPL 的规定，哪些情况不算违背 GPL 的规定，这就意味着你很有可能需要法律部门介入。”在 Jagielski 看来，Apache 的许可条款就简单易懂得多，没有接受过专门的法律培训的人也能看得明白。“这些条款非常容易理解，因此，企业如果采用其风险就会小得多。”尽管许可条款非常关键，但是，开源软件开发者还必须关注其他问题，比如，各个许可对应的软件背后的开发社区的深度和广度。“之所以这些许可类型得到比较多的公司的认可，就是因为背后的社区足够成熟。”Lindberg 律师说。

由于历史悠久而且在市场上占有绝对的优势，GPL 有一个庞大、深入而且活跃的开发社区，不过其他的许可类型正在快速跟上。“我想我们会看到其他许可类型的社区越来越活跃，它们也将向我们证明这些许可类型也可以建立和维护一个不弱于 GPL 的社区。”Lindberg 说，“这将给那些偏爱其他的许可（如 Apache、Eclipse 等）的人更多机会，可以根据自己需要来选择最合适的软件许可方式。”

GPL 妨碍商业云应用的发展

为了保证源代码的自由分发，GPL 要求采用 GPL 许可方式发布的软件必须把源代码同时发布。“根据 GPL 的规定，采用 GPL 的许可意味着你发布的软件其他人可以自由地使用，同时，源代码人

们也可以自由地查看、拷贝。”分析师 Kiewe 说。

但是，云计算的出现使得这一 GPL 的规定出现了漏洞。因为软件的运行不一定需要发布，比如 SaaS 就可以通过互联网交付给使用者，使用者只需要通过互联网访问发布者的服务器即可得到所需的服务，而无需发布软件和软件源代码。这就意味着，在 GPL 许可之下的云应用根本就不需要发布源代码。因此，很多云应用的开发者尽管采用的是 GPL 许可，却不再严格执行发布源代码的规定。

“传统的 GPL 许可对于 Salesforce.com 这样的 SaaS 运营商以及 Google 这样的搜索引擎供应商而言根本就不适用。”Lindberg 说，这一事情已经引起了自由软件基金（Free Software Foundation）组织的注意，SaaS 漏洞有望在名为 Affero 的新版 GPL 中得以解决。“为了保证在新的计算环境下 GPL 原来的精神得到坚守，这是必要的。”Lindberg 说。

然而，这一行动将迫使云供应商必须公开它们的软件源代码，这显然不是这些云供应商所能接受的，所以，它们不得不寻求新的软件许可，其结果就是让 GPL 及其其精神远离了云计算。Eclipse 的 Milinkovich 介绍说，Amazon.com 为了避免公开源代码的要求，其弹性计算云（Elastic Computing Cloud，EC2）服务采用的就是 Eclipse 许可。

新的软件授权方式不断涌现

那些认为 GPL 过于苛刻、会影响企业经营的人其实还有很多其他的选择，Apache、Eclipse 就是其中的两个。Apache 凭借其在 Web 开发领域的影响力而受到了部分 Web 应用开发者的喜欢，也是 GPL 主要的竞争对手之一；Eclipse 最初只是一个面向 Java 开发者的开发工具项目，而现在已经扩展到很多软件领域。其他比较重要许可方式还有 Artistic、BSD、MIT、Mozilla Public License (MPL)等以及其他数十个应用范围更窄一些许可方式。

“我相信随着新的商业模式不断涌现，软件授权许可方式也必然要不断创新，这对整个行业的发展来说非常必要的，也有着积极意义。”开源软件开发者同时也是 Openbravo COO 的 Josep Mitja 说，“不过，许可方式的创新也带来问题，许可方式太多了，供应商们常常不知道哪种方式更适合自己，有时供应商需要花费几周甚至几个月才能找到一个最适合自己的授权许可方式。”Josep Mitja 说：“从整个行业发展的角度来看，也需要防止许可方式无限制滋生，这会给市场带来混乱。”

GPL 的作用不应低估

尽管面临着很多更新的、更有利于企业开展经营活动的软件许可规范的竞争，但是，GPL 仍然还有存在的价值，也并不能说它就面临被抛弃的命运。“在整个开源领域，绝大多数代码采用的是 GPL 许可方式。”观察家 Lindberg 律师说，“这就是 GPL 的最大优势，也是 GPL 的价值所在。”

分析师 Kiewe 认为，不管是 GPL 还是 GPL 的竞争对手们未来都会和平共处，各自在不同的场合发挥自己的作用。Kiewe 相信，对于那些不以赢利为目的人而言，GPL 仍然是最好的许可方式。“在这个世界上总有很多理想主义者，他们希望与人分享自己的东西，而不希望这些东西在某种许可方式下封闭起来，使得此后所有的开发都变成一种商业行为。”“如果你在大学或者某个研究单位工作，你希望把你的一些想法与所有人分享，同时你希望别人能自由地使用你的这些代码，而不期望有人盗用你的思想把它们用于商业目的，那么，GPL 许可是非常适宜的。”Kiewe 说。

不过，Lindberg 预计，随着开源市场进一步扩大和市场的进一步细分，GPL 目前的统治地位一定会逐渐改变。“最终我们肯定会看到，市场上多种开源软件的许可方式共存，每一种都有自己适用的行业和适用的具体条件，谁也不能取代谁。”

技术沙龙

****

2009 系统架构师大会精彩回眸

2009 年 8 月 28 日至 29 日，北京歌华开元饭店，2009 系统架构师大会（SACC2009）盛大召开。本次系统架构师大会由 IT168、 ChinaUnix、ITPUB、IXPUB 共同主办。这场盛会汇集了系统架构及相关领域中的众多专家和学者，很多嘉宾都是来至知名企业的技术一线，有着丰富的领域经验和成功实践，并且，最宝贵的是，很多特约嘉宾都带来了各自在多年从业经历中总结出的优秀实践材料，到这里与各位业内朋友们分享。

本次大会主要分为第一天上午的系统架构整体设计、第一天下午的网络架构设计和应用服务器架构设计、第二天上午的开发数据库架构设计和系统安全监控，以及第二天下午的存储架构设计和教育行业架构设计等共计七个讨论会场。

以下为部分精彩会场内容节选：更多的内容可以进入本次大会的专题：

http://linux.chinaunix.net/bbs/forumdisplay.php?fid=24&filter=type&typeid=55

LVS（Linux Vitual Server）在行动

IT168 记者：琰珺

组建高可扩展的网络服务是一项非常复杂的工程，而且花费高昂。对此，LVS（Linux Vitual Server）源代码研发者、TelTel 的首席科学家章文嵩主张："LVS 可以使得这项工作（组建高可扩展的网络服务）变得容易起来，而且 LVS 已经被 证明非常稳定，也正在被越来越多站点和系统所部署。"

章文嵩的演讲稿

Ok，基于开源的项目开发已然成为国内众多系统架构师或编程人员的参照首选，然而 LVS 作为开源性质的系统架构及部署则再一次将 Linux 的精神推向了高 潮。在 8 月 28 日-29 日召开的 2009 年系统架构师大会上，章文嵩作为网络架构设计专场的首场演讲人得到了原本 400 人会场的爆满人气，

这或许已经说明 了 LVS 的魅力。

然而，究竟人们为什么对组建高可扩展的网络服务有如此高的关注度呢？正如章文嵩所阐述的那样，随着 Internet 的飞速发展，横行在网络上的流量以每年 100% 速度增长，各行业或企业都有很多关键性业务在网上开展。由此所带来的直接结果就是：很多网站收到前所未有的工作负载，更别说如今视频分享网站、社交网络网站等新兴互联网应用的接连出现的风靡……或许这些描述还不足以明确高可扩展的网络服务组建需求有多么紧迫，那么诸如系统平台所需要的渐进的可扩展性、7*24 小时的可用性、可管理性、价格有效性等非常硬性的技术需求则足以清晰。

对于上述需求的解决之道，章文嵩指出，可能的解决方案有 2 种：一种是，针对单服务的升级，但是升级过程复杂、成本高、且针对的往往是单一故障点；另一种是，针对服务器集群的升级，也即架设网络服务的有效结构，这样做的好处在于：通过增强集群系统的冗余性从而实现高可用性、通过分而治之实现高性能和高吞吐率、通过对结点数目的动态调整实现高可扩展性和高性能/价格比等。

简要来讲，章文嵩给出了 IP Virtual Server 的实现要素：1.在 Linux 内核中实现；2.三种 IP 负载均衡技术（通过网络地址转换实现虚拟服务器、 通过 IP 隧道实现虚拟服务器、通过直接路由实现虚拟服务器）；3.十种负载调度算法；4.支持 IPv4 和 IPv6。对于系统的高可用性而言，可组建的集群可以是 WEB 集群、Cache 集群、邮件集群、Media 集群、DNS 集群、MySql 集群等。此外，LVS 的硬件平台方面：1.任何跑 Linux 的硬件 平台都可以运行 LVS；2.LVS 的负载均衡和报文转发对 CPU 速度要求低；3.可以用低功耗的硬件平台运行 LVS，如 Intel ATOM CPU 1.6GHz, 功耗 2~3W，及 Gigabit Ethernet。

与此同时，我们不得不承认，Linux 开源项目给广大的系统架构师提供了非常好的资源平台，更值得向往的是，一旦这个平台得到全球系统架构师智囊团级别的全力支撑时，LVS 项目更能为所有热衷于网络应用的人们带来最贴心的安慰。

应用 MySQL 数据库进行在线灾备实践案例

IT168 记者：李隽

在田逸的演讲结束后，51.com 数据库主管徐景春也向大家介绍了 51.com 在在线灾备方面的探索和研究。

徐景春介绍说，51.com 公司从管理架构上就极为关注数据安全，因此身为数据库主管，徐景春本人在工作中也一直谨慎小心，从各种角度保障数据库的备份与灾备。而与此同时，一则报纸上的热点消息也更让 51.com 在灾备方面提高警惕：

去年，德克萨斯某公司丢失 27GB 档案数据没有备份，给该公司业务造成严重损失，为此，负责该备份项目的 IBM 公司被判罚 90 万美金。根据徐景春讲述， 当时该公司老总以这例新闻提醒公司技术管理人员提高备份方面的安全措施，并要求彻底检查公司的备份措施是否完备。可见数据安全对于 51.com 的重要性。

提高数据安全性可能包括两个方面，首先是备份，其次是容灾，包括同城容灾和异地容灾，对于中国的大部分互联网公司来说，容灾方案实际上还是一个比较奢侈 的方案，为此，徐景春详细介绍了 51.com 在备份方面的探索，实际上，备份也是灾备的重要组成部分，一个完整的备份方案已经能够应付大多数系统计划内和 计划外的宕机情况。

最早 51.com 在成立初期，只有几台 DB 服务器，相互镜像，不要说灾备，普通的备份安全标准

也完全没办法保证。一般来说都是通过手工输入命令进行备份。随着 51.com 业务发展，前端服务器数量迅速发展为几百台甚至上千台，此时，他们采用了自写的脚本进行自动化备份。但此时也同样产生一些问题，例如自写的脚本通常需要维护一个设备列表，当需要备份的服务器数量过于庞大的时候，通过认为操作这个列表变得极其繁琐，且出错率高。

51.com 数据库主管徐景春

而且随着公司业务模式的发展，备份需求也发生了较大的分化，某些应用例如交易结算型要求每小时就进行一次备份，某些应用则需要一天进行一次备份，而某些 不重要的数据一个星期做一次备份即可。这个时候一套备份系统已经没办法满足需求了。因此需要为整个公司所有的备份需求进行统一的规划和整理。

这个时候 51.com 的备份体系已经非常复杂，Oracle 备份与 MySQL 备份并存，手工备份与自动化备份并存，增量备份与全量备份混合使用，同时， 该公司还实现了对电信和网通 IDC 机房的交叉同城异地备份，例如周一数据异地备份到电信机房，周二数据异地备份到网通机房，这样无论任何一边的机房挂 掉，51.com 整体系统仍然能尽可能多的数据，已经拥有一定的抗灾难能力。

技术新知

****

独辟蹊径 J2EE 网站之 Tomcat 篇(上)

----基于 mod_jk 整合 apache 与 tomcat

ChinaUnix 网友：gamester88、kns1024wh、loveradmin

本文介绍在 centos 5.x 环境下通过 yum 源的扩展使用 munin, monit,ntop 工具来监管你的应用程序和服务器。题为懒人说说的是简便的安装方式而已，将强大的功能配置简单的应用起来是很

尽管 LAMP 任然是很多 WEB 站点的主流，不过随着开源业界的收购风波引发大家更多的关注 J2EE 的企业应用。进而很多的 Linux 系统管理员也需要掌握 J2EE 的开发平台、测试环境、运行环境的搭建和管理。文本将通过在 CentOS 5.2 上通过 mod_jk 整合 Apache 2.2、JDK6.13、Tomcat 6.0 并通过 mysql-connector-java 实现 MySQL 5.0 的数据存储，以透明化 Java 应用的 Bundle 版本的构建方式，并介绍如何通过 jsvc 实现 Tomcat 的开机启动方式，通过 cronolog 分割日志并使用 awstats 可视化分析，以及优化，通过 Apache Tomcat 虚拟主机结合的方式实现主机资源的最大化的利用率。

Apache 是个性能很强的 web 服务器，而 Tomcat 是开源的 jsp 容器。在实际的 j2ee 项目部署中，可以很好的利用两者的特性，将 apache 和 tomcat 整合起来，主要的好处为将 apache 作为服务器，将静态页面的处理交给它，很好的利用了 apache，提高系统的整体性能；可以通过一些连接技术手段，将 jsp 请求转发到多个 tomcat 容器，可以做到负载均衡。

为便于 CentOS 5.2 系统的组件的安装及管理首先在 CentOS 5.2 的系统上构建一个本地的 yum 源，然后采用源码的方式安装 apache-tomcat-6.0.18.tar.gz、jaf-1_0_2-upd2.zip、jdk-6u13-linux-i586.bin、tomcat-connectors-1.2.28-src.tar.gz、httpd-2.2.11.tar.bz2、javamail-1.4.2.zip、mysql-connector-java-5.1.7.tar.gz.tar、aribIk.rar、cronolog-1.6.2.tar.tar、awstats-6.9-1.noarch.rpm、jsvc.tar.gz 等，所用到源码包在其官方站点或者是搜索引擎上都是完全可以搜索到。

以上所有软件均可以在 download.chinaunix.net 网站获得最新版本。

本文的演示系统说明，系统为 Centos 5.2，安装基本的 core 包，IP 地址 192.168.0.253，主机名为 tomcat，配置过程中使用的域名为 testdemo.com。

一、使用 ISO 介质搭建本地 yum 源

通过 yum 可以便捷的安装软件包同时可以很好的解决软件依赖关系，并自动安装需要依赖的软件包，使用互联网上的源如果网络的质量不是很好，将是安装延迟很多；建立本地主机的 yum 源可以在瞬间完成必要的软件包的安装。

a)挂在光盘介质

```
[root@tomcat ~]# mount /dev/cdrom /mnt
mount: block device /dev/cdrom is write-protected, mounting read-only
```

b)将光盘中的文件复制到本地/usr/src/source 中，使用 tar 命令完成

```
[root@tomcat ~]#tar  – cvf - /mnt | tar  – xvf - -C /usr/src/source
```

c)编辑本地 yum 的 repo 文件

```
[root@tomcat ~]#rpm  – rf /etc/yum.repos.d/*
```

将默认的 yum 源删除，建立新的本地源配置文件 local.repo

```
[root@tomcat ~]# vi /etc/yum.repos.d/local.repo
[local]
name=CentOS 5 Local Repository
baseurl=file:///usr/src/source/
enabled=1
gpgcheck=0
[root@tomcat ~]#yum clean all
```

清空当前的 yum 缓存信息，并使用 yum 安装必要的 gcc 等必要的编译工具包

```
[root@tomcat ~]# yum install gcc
[root@tomcat ~]# yum install openssl*
```

二、安装 Apache

安装 apache 的源码包，这里要有一个好的就是将需要编译的源码包放置在/usr/local/src 目录下面。

```
[root@tomcat ~]# cd /usr/local/src/
[root@tomcat src]# ls
apache-tomcat-6.0.18.tar.gz  jaf-1_0_2-upd2.zip  jdk-6u13-linux-i586.bin
tomcat-connectors-1.2.28-src.tar.gz   httpd-2.2.11.tar.bz2       javamail-1.4.2.zip
mysql-connector-java-5.1.7.tar.gz.tar
```

源码编译 apache ，本文中使用的版本是 httpd-2.2.11.tar.bz2，源码编译的三步 configure 结合适当的参数，然后就是执行 make 和 make install 如果没有出现 error 的错误就可以正常的使用了。

```
[root@tomcat src]# tar jxvf httpd-2.2.11.tar.bz2
[root@tomcat src]# cd httpd-2.2.11
[root@tomcat httpd-2.2.11]# ./configure --prefix=/usr/local/httpd
--with-mpm=worker --enable-cache --enable-file-cache
--enable-disk-cache --enable-mem-cache --enable-mime-magic --enable-
headers
--enable-ssl --enable-http --enable-cgi --enable-rewrite --enable-so
```

```
--with-suexec-gidmin --with-suexec-logfile
[root@tomcat httpd-2.2.11]# make
[root@tomcat httpd-2.2.11]# make install
```

安装软件包后要设置 apache 的启动服务脚本，简单的方式就是将 apachectl 复制到 init.d 目录下面。

```
[root@tomcat httpd-2.2.11]# cp /usr/local/httpd/bin/apachectl
/etc/rc.d/init.d/apache
[root@tomcat httpd-2.2.11]# /etc/rc.d/init.d/apache start
httpd: apr_sockaddr_info_get() failed for tomcat
httpd: Could not reliably determine the server's fully qualified domain name,
using 127.0.0.1 for ServerName
```

出现这个提示信息说明的是 apache 的配置文件中的 ServerName 开关项没有设置完成的 FQDN 名称，可以简单的修改为 localhost 或本机的 IP 地址。

```
[root@tomcat httpd-2.2.11]# vi /usr/local/httpd/conf/httpd.conf
ServerName localhost:80
```

重新启动 apache 服务，并查看进程及端口信息

```
[root@tomcat httpd-2.2.11]# /etc/rc.d/init.d/apache stop
[root@tomcat httpd-2.2.11]# /etc/rc.d/init.d/apache start
[root@tomcat httpd-2.2.11]# ps aux | grep httpd; netstat -ntulp | grep :80
root     29285  0.5  0.4   7420  2476 ?        Ss   17:20   0:00
/usr/local/httpd/bin/httpd -k start
daemon   29286  0.0  0.3   7192  1564 ?        S    17:20   0:00
/usr/local/httpd/bin/httpd -k start
daemon   29287  0.0  0.3 284056  1936 ?        Sl   17:20   0:00 /usr/local/httpd/bin/
httpd -k start
daemon   29289  0.0  0.3 284056  1940 ?        Sl   17:20   0:00 /usr/local/httpd/bin/
httpd -k start
daemon   29291  0.0  0.3 284056  1940 ?        Sl   17:20   0:00 /usr/local/httpd/bin/
httpd -k start
root     29372  0.0  0.1   3904   672 pts/0    R+   17:20   0:00 grep httpd
tcp        0      0 :::80                   :::*                    LISTEN      29285/httpd
```

三、安装 Tomcat 配置 mod_jk 实现和 apache 的整合

安装 tomcat 是相对很简单的，需要做的是首先配置好 tomcat 运行的 JDK 的环境，这个主要是配置 profile 中的环境变量。这里还是提示一个良好的文件夹的组织结构，对于 JAVA 非 Linux 系统自

带的软件包建议放置在/opt 或者/usr/local/文件夹下面，本文放置在/opt 目录下面。

```
[root@tomcat httpd-2.2.11]#mkdir /opt;cd /usr/local/src
#创建/opt 文件夹，或者在磁盘分区时挂在上文件系统
[root@tomcat src]# mv apache-tomcat-6.0.18.tar.gz tomcat-connectors-1.2.28-src.tar.gz jdk-6u13-linux-i586.bin jaf-1_0_2-upd2.zip javamail-1.4.2.zip  /opt
#将 jdk、tomcat、 mod_jk 等移动到/opt 文件夹中
[root@tomcat opt]# tar zxvf apache-tomcat-6.0.18.tar.gz ; unzip jaf-1_0_2-upd2.zip ; unzip javamail-1.4.2.zip ; tar zxvf tomcat-connectors-1.2.28-src.tar.gz
#解压软件包
```

安装配置 JDK 运行环境

```
[root@tomcat opt]#chmod +x jdk-6u13-linux-i586.bin
#为 jdk 安装包添加执行权限
[root@tomcat opt]# ./jdk-6u13-linux-i586.bin
Sun Microsystems, Inc. Binary Code License Agreement
for the JAVA SE DEVELOPMENT KIT (JDK), VERSION 6
~~~~~~~~~~~~~~~~~~ignore~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
省略软件包的协议信息
Do you agree to the above license terms? [yes or no]
yes
~~~~~~~~~~~~~~~~~~~~ignore~~~~~~~~~~~~~~~~~~~~~~``
省略软件包的安装信息
http://java.sun.com/javase/registration/JDKRegistrationPrivacy.html
Press Enter to continue.....
Done.
#执行 JDK 软件包的安装，安装提示确认许可协议等信息
```

为软件包建立软连接，软连接是可以方便软件版本管理的有效方式，当有新的软件包需要更新的时候需要做的是调整 ln 建立新的软连接，这样可以同时做到以前版本的备份；当然也可以使用 mv 直接重命名软件包的文件夹。

```
[root@tomcat opt]#ln –s mv apache-tomcat-6.0.18 tomcat; ln –s  jdk1.6.0_13 java; ln –s  jaf-1.0.2 jaf; ln –s  javamail-1.4.2 javamail
#使用 ln 建立软连接的方式
[root@tomcat opt]# mv apache-tomcat-6.0.18 tomcat; mv jdk1.6.0_13 java;
```

```
mv jaf-1.0.2 jaf; mv javamail-1.4.2 javamail
#使用 mv 重命令文件夹的方式
```

编辑/etc/profile 文件设置 JAVA 运行的环境变量

```
[root@tomcat opt]# vi /etc/profile
JAVA_HOME=/opt/java
CATALINA_HOME=/opt/tomcat
CATALINA_BASE=/opt/tomcat
CLASSPATH=./:${JAVA_HOME}/lib:${JAVA_HOME}/jre/lib/ext:${CATALINA_HOME}/common/lib:/opt/javamail/mail.jar:/opt/jaf/activation.jar
export JAVA_HOME CATALINA_HOME CATALINA_BASE CLASSPATH
export PATH=${JAVA_HOME}/bin:${CATALINA_HOME}/bin:${PATH}
[root@tomcat opt]# source /etc/profile
#可以重新启动系统是配置的环境便利生效，也可以使用 source 是配置的环境变量生效
[root@tomcat opt]# java  – version
#执行此命令检查设置的环境变量是否已经生效
```

为 apache 和 tomcat 的整合编译 mod_jk 模块

```
[root@tomcat opt]# cd tomcat-connectors-1.2.28-src/native/
[root@tomcat native]# ./configure --with-java-home=/opt/java--with-apxs=/usr/local/httpd/bin/apxs
[root@tomcat native]# make
#至此 mod_jk 模块编译完成
```

将 mod_jk 模块安装到 httpd 中，就是将编译好的 mod_jk.so 复制到 apache 的 modules 文件夹中或者是 lib 文件夹中，本文是放置在 lib 文件夹中。

```
[root@tomcat native]# cp apache-2.0/mod_jk.so /usr/local/httpd/lib/
```

编辑 apache 的配置文件加载启用 mod_jk 模块

```
[root@tomcat opt]# vi /usr/local/httpd/conf/httpd.conf
LoadModule jk_module lib/mod_jk.so
# mod_jk settings
Include conf/mod_jk.conf
#在配置文件中 LoadModule 末尾加入上面内容,载入 mod_jk 模块与 mod_jk 配置文件
```

在 apache 配置文件夹中编辑 mod_jk 配置文件，在此文件中配置那些网站的文件或者是网站的目录需要交由 servelet 来执行解析。

```
[root@tomcat opt]# cd /usr/local/httpd/conf/
[root@tomcat conf]# vi mod_jk.conf
#setup the workers.properties file path, default prefix path is httpd's home (/usr/local/httpd)
JkWorkersFile conf/workers.properties
JkLogFile /var/log/jk.log
JkShmFile /var/log/jk-runtime-status
JkLogLevel error
JkLogStampFormat "[%a %b %d %H:%M:%S %Y] "
JkOptions +ForwardKeySize +ForwardURICompat -ForwardDirectories
JkRequestLogFormat "%w %V %T"
# Sample JkMounts. Replace these with the paths you would
# like to mount from your JSP server.
# syntax: JkMount ${URL_DIR}/*.jsp worker_name
JkMount /project1/*.jsp jsp-ap207
JkMount /servlet/* jsp-ap207
JkMount /*.jsp jsp-ap207
JkMount /*.do jsp-ap207
JkMount /*.action jsp-ap207
JkMount /*.java jsp-ap207
JkMount /authImg jsp-ap207
JkMount /fckeditor/* jsp-ap207
JkMount /*.act  jsp-ap207
#具体要配置那些网站的访问解析目录交由 servelet 程序执行，可以与开发人员进行沟通或者通过 apache 的 error 日志来获取必要的信息，如在 apache 日志的 error 文件中看到有 File does not exist: /var/www/html/osb-theme, referer: http://192.168.1.148/web/guest/home；File does not exist: /var/www/html/dwr, referer: http://192.168.1.148/group/dongyou/emergency 则需要将对用的网站目录的解析交由 jsp-ap207 这个 servelet 程序执行解析。
```

编辑 Tomcat 的链接器（Connector）文件 workers.properties，此文件定义 servelet 信息

```
[root@tomcat opt]# cd /usr/local/httpd/conf/
```

```
[root@tomcat conf]# vi workers.properties
# BEGIN workers.properties
# setting tomcat_home and java_home
workers.tomcat_home=/opt/tomcat
workers.java_home=/opt/java
# worker.list defined worker_name, used by mod_jk.conf
worker.list=jsp-ap207
worker.jsp-ap207.port=8009
worker.jsp-ap207.host=192.168.0.253
worker.jsp-ap207.type=ajp13
worker.jsp-ap207.lbfactor=1
#当启动服务器的时候，Web 服务器插件会把这些出现在 worker.list 属性中出现名字的 worker 实例化，而这些也就是你可以用来映射请求的 worker。
#ajp13 这种 worker 知道如何使用 ajpv13 协议去给用来外部处理的 worker 传递一个请求。
#负载平衡 Worker（lb Worker）属性，负载平衡 worker 并不是真正的与 Tomcat 的其它 worker 通信，而是负责对若干“真实”的 workers 的管理，这个也是配置 Tomcat 应用群集的基础
```

这个 Workers 实际上属于 Tomcat 的链接器（Connector），代表了一个 Tomcat 实例，这个实例代表了由某种 web 服务器来执行 servelet 程序。例如 apache 来把 servelet 请求转递 Tomcat 进程（worker）来进行后台处理。

设置 tomcat 站点的运行信息，配置 server.xml 文件

```
[root@tomcat conf]# cd /opt/
[root@tomcat opt]# vi tomcat/conf/server.xml
<Host name="192.168.0.253" debug="0" appBase="/default"
unpackWARs="true" autoDeploy="true">
<Context path="/project1" docBase="/projects/project1"
debug="5" reloadable="true" crossContext="true">
 </Context>
 </Host>
#上面的部分定义 tomcat 的 appBase 和 docBase 信息
appBase 表示这个目录下面的子目录将自动被部署为应用或者 这个目录下面的.war 文件将被自动解压缩并部署为应用；
```

docBase 只是指向了某个应用的目录；

就是说如果你想自己指定路径，那么应该在 docBase 里面，如果你想简单，那么直接把他们复制到 appBase 下面就行了

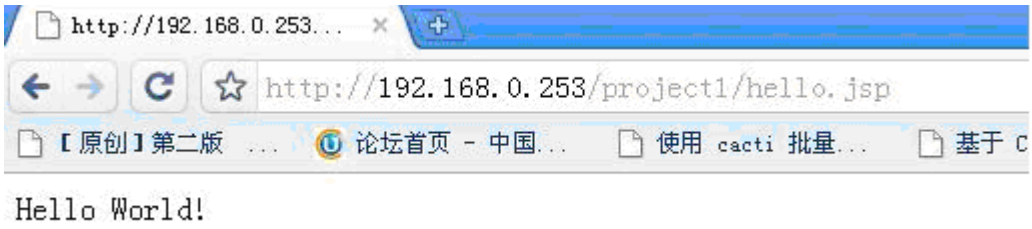
建立 docBase 和 appBase 文件夹，并在其中建立 hello.jsp 测试文件

```
[root@tomcat opt]# mkdir -p "/projects/project1"; cd /projects/project1
[root@tomcat project1]# vi hello.jsp
<%@ page contentType="text/html;charset=big5" %>
<%
String str1="Hello World!";
out.println(str1);
%>
```

重新启动 tomcat 服务，并访问测试页面

```
[root@tomcat project1]# /etc/rc.d/init.d/apache stop
[root@tomcat project1]# /etc/rc.d/init.d/apache start
[root@tomcat project1]# /opt/tomcat/bin/startup.sh
Using CATALINA_BASE:   /opt/tomcat
Using CATALINA_HOME:   /opt/tomcat
Using CATALINA_TMPDIR: /opt/tomcat/temp
Using JRE_HOME:        /opt/java
```

访问测试页面信息



实际应用中 apache 与 tomcat 的整合有 mod_jk 模块、apache 的 http_proxy、apache 的 ajp_proxy 几种方式。apache 的 http_proxy 方式是利用 Apache 自带的 mod_proxy 模块使用代理技术来连接 Tomcat；ajp_proxy 连接方式其实跟 http_proxy 方式一样，都是由 mod_proxy 所提供的功能。配置也是一样，只需要把 http:// 换成 ajp:// ，同时连接的是 Tomcat 的 AJP Connector 所在的端口。相对于 mod_jk 的连接方式，后两种在配置上是比较简单的，灵活性方面也一点都不逊色。

ajp_proxy 配置的示例参考

```
<IfModule mod_proxy.c>
```

```
ProxyPass       / ajp://localhost:8009/
ProxyPassReverse  / ajp://localhost:8009/
</IfModule>
```

四、MySQL 配置部分

开源的 MySQL 可以作为很好的后端数据库存储数据库，J2EE 的应用与 MySQL 数据库链接需要使用 jdbc（mysql-connector-java）连接，本文中使用 mysql-connector-java-5.1.7.这个版本，MySQL 使用构建的本地的 yum 源进行安装。

通过本地 yum 安装 MySQL Server，设置 mysql 的 root 用户密码，不推荐使用 mysql 的 root 用户为空密码。

```
[root@tomcat project1]# yum install mysql*
#通过 yum 命令安装 MySQL Server
[root@tomcat project1]# /etc/rc.d/init.d/mysqld star
#启动 MySQL 服务
[root@tomcat project1]# mysqladmin -u root -p password chinaunix
Enter password:
#通过 mysqladmin 命令设置 root 用户的新密码为 chinaunix；mysql 默认没有密码。因为以前没有密码，直接回车。
[root@tomcat project1]# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.0.45 Source distribution
#使用新密码登陆 MySQL
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
mysql> CREATE DATABASE project1;
Query OK, 1 row affected (0.00 sec)
#创建 project1 数据库
mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
```

```
| mysql              |
| project1           |
| test               |
+--------------------+
4 rows in set (0.00 sec)
mysql> GRANT ALL PRIVILEGES ON *.* TO test@”%”
    IDENTIFIED BY ‘test’ WITH GRANT OPTION;
#为 tomcat 连接 MySQL 数据库创建 test 用户，可以从任何地方连接服务器的一个完全的超级用户，但是必须使用一个口令(‘test’)做这个。注意，必须对 test@localhost 和 test@”%”发出 GRANT 语句。
mysql> flush privileges;
#重新加载新的权限表
```

安装 jdbc（mysql-connector-java）连接器不需要编译，只需要将相应的 jar 包复制到 tomcat 的 lib 文件夹中，然后配置对用的链接参数信息。

```
[root@tomcat project1]# cd /usr/local/src/
[root@tomcat src]# tar xvf mysql-connector-java-5.1.7.tar.gz.tar
[root@tomcat src]# cp mysql-connector-java-5.1.7/mysql-connector-java-5.1.7-bin.jar /opt/tomcat/lib/
#完成 jdbc（mysql-connector-java）连接器的安装
```

配置 tomcat 的 MySQL 数据库的连接信息

```
[root@tomcat src]# cd /opt/
[root@tomcat opt]# vi tomcat/conf/server.xml
<Host name="192.168.0.253" debug="0" appBase="/default" unpackWARs="true" autoDeploy="true">
<Context path="/project1" docBase="/projects/project1" debug="5" reloadable="true" crossContext="true">
<Resource name="jdbc/project1" auth="Container" type="javax.sql.DataSource" maxActive="100" maxIdle="30" maxWait="10000" username="test" password="test" driverClassName="com.mysql.jdbc.Driver" rl="jdbc:mysql://127.0.0.1:3306/project1?autoReconnect=true"/>
</Context>
</Host>
#设置 jdbc 的连接参数
```

建立 db_test.jsp 测试文件，测试 mysql 的链接

```
[root@tomcat opt]# vi /projects/project1/db_test.jsp
<%@ page import="java.sql.*"%>
<%@ page import="javax.sql.*"%>
<%@ page import="javax.naming.*"%>
<%@ page session="false" %>
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=big5">
<title>Test of mysql connection pool</title>
</head>
<body>
<%
out.print("start<br/>");
try{
Context initctx = new InitialContext();
Context ctx = (Context) initctx.lookup("java:comp/env");
Object obj = (Object) ctx.lookup("jdbc/project1");
javax.sql.DataSource ds = (javax.sql.DataSource)obj;
Connection conn = ds.getConnection();
out.print("mysql connection pool runs perfectly!");
conn.close();
}
catch(Exception ex){
out.print(ex.getMessage());
ex.printStackTrace();
}
%>
</body>
</html>
```

重新加载 tomcat 服务

```
[root@tomcat opt]# /opt/tomcat/bin/shutdown.sh
Using CATALINA_BASE:   /opt/tomcat
Using CATALINA_HOME:   /opt/tomcat
Using CATALINA_TMPDIR: /opt/tomcat/temp
Using JRE_HOME:        /opt/java
[root@tomcat opt]# /opt/tomcat/bin/startup.sh
Using CATALINA_BASE:   /opt/tomcat
Using CATALINA_HOME:   /opt/tomcat
Using CATALINA_TMPDIR: /opt/tomcat/temp
Using JRE_HOME:        /opt/java
```

通过浏览器访问 db_test.jsp 测试页面


```
start
mysql connection pool runs perfectly!
```

五、Apache 和 Tomcat 服务启动设置

a)为源码编译的 apache 添加启动服务脚本

第一种方法:使用 rc.local 脚本

在/etc/rc.d/rc.local 中添加启动 apache 服务的命令

```
[root@tomcat ~]# vi /etc/rc.local
/usr/local/httpd/bin/apachectl start
```

第二种方法：设置一个标准的启动服务的脚本，并放置在/etc/rc.d/init.d 下面

```
[root@tomcat ~]# cp /usr/local/httpd/bin/apachectl /etc/rc.d/init.d/apache
#将 apachectl 复制为/etc/rc.d/init.d/apache 脚本名称
ln -s /etc/init.d/apache /etc/rc1.d/K90apache
ln -s /etc/init.d/apache /etc/rc2.d/S90apache
ln -s /etc/init.d/apache /etc/rc3.d/S90apache
ln -s /etc/init.d/apache /etc/rc4.d/S90apache
ln -s /etc/init.d/apache /etc/rc5.d/S90apache
ln -s /etc/init.d/apache /etc/rc6.d/K95apache
```

```
#建立不同运行级别的脚本链接
[root@tomcat ~]# vi /etc/rc.d/init.d/apache
 #!/bin/sh:
 #chkconfig:    2345  85  15
 #description:  apache starup
#在此文件的最前面添加上面的启动服务信息
[root@tomcat ~]# chkconfig --level 345 apache on
#将 apache 添加为启动服务
```

然后 Linux 启动以看到 apache 服务已经自动运行

b)为 tomcat 开机自动启动服务脚本

使用 jsvc 将 tomcat 启动为 Linux 的一个进程

```
[root@tomcat ~]# cd /opt/tomcat/bin/
[root@tomcat bin]# tar zxvf jsvc.tar.gz
[root@tomcat bin]# cd jsvc-src/
[root@tomcat jsvc-src]# chmod +x configure
[root@tomcat jsvc-src]# ./configure --with-java=/opt/java
[root@tomcat jsvc-src]# make
#完成 jsvc 的编译
[root@tomcat jsvc-src]#cp /opt/tomcat/bin/jsvc-src/native/Tomcat5.sh
/etc/rc.d/init.d/
#将生成的脚本复制到/etc/rc.d/init.d 文件中
[root@tomcat jsvc-src]# cd /etc/rc.d/init.d/
[root@tomcat init.d]# chmod +x Tomcat5.sh
[root@tomcat init.d]# vi Tomcat5.sh
JAVA_HOME=/opt/java
CATALINA_HOME=/opt/tomcat
DAEMON_HOME=/opt/tomcat
TOMCAT_USER=root
TMP_DIR=/var/tmp
PID_FILE=/var/run/jsvc.pid
CATALINA_BASE=/opt/tomcat
$DAEMON_HOME/bin/jsvc-src/jsvc \
注意 Tomcat5.sh 里的一些路径配置
[root@tomcat init.d]# ln -s /etc/init.d/tomcat5.sh /etc/rc0.d/K90tomcat5.sh
[root@tomcat init.d]# ln -s /etc/init.d/tomcat5.sh /etc/rc1.d/K90tomcat5.sh
[root@tomcat init.d]# ln -s /etc/init.d/tomcat5.sh /etc/rc2.d/S90tomcat5.sh
```

```
[root@tomcat init.d]# ln -s /etc/init.d/tomcat5.sh /etc/rc3.d/S90tomcat5.sh
[root@tomcat init.d]# ln -s /etc/init.d/tomcat5.sh /etc/rc4.d/S90tomcat5.sh
[root@tomcat init.d]# ln -s /etc/init.d/tomcat5.sh /etc/rc5.d/S90tomcat5.sh
[root@tomcat init.d]# ln -s /etc/init.d/tomcat5.sh /etc/rc6.d/K95tomcat5.sh

#为不同的运行基本建立启动和停止服务信息
```

测试 tomcat 启动脚本

```
[root@tomcat init.d]# /etc/rc.d/init.d/Tomcat5.sh start

#执行脚本启动 tomcat 服务

[root@tomcat init.d]# netstat -tnl
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address Foreign Address State
tcp 0 0 0.0.0.0:3306 0.0.0.0:* LISTEN
tcp 0 0 :::8009 :::* LISTEN
tcp 0 0 :::8080 :::* LISTEN
tcp 0 0 :::80 :::* LISTEN
tcp 0 0 :::22 :::* LISTEN

#查看服务状态
[root@tomcat init.d]# /etc/rc.d/init.d/Tomcat5.sh stop
[root@tomcat init.d]# netstat -tnl
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address Foreign Address State
tcp 0 0 0.0.0.0:3306 0.0.0.0:* LISTEN
tcp 0 0 :::80 :::* LISTEN
tcp 0 0 :::22 :::* LISTEN
#检测服务是否已经停止
```

为 tomcat 启动脚本执行用户为 tomcatadmin，不以 root 用户身份执行此脚本

```
[root@tomcat init.d]# useradd tomcatadmin
#为 tomcat 启动脚本增加一个执行用户 tomcatadmin 不使用 root 用户身份执行

[root@tomcat init.d]# vi Tomcat5.sh
TOMCAT_USER=tomcatadmin

#将 tomcat 运行用户的身份更改为 tomcatadmin
[root@tomcat /]# chown -R tomcatadmin:tomcatadmin /opt/tomcat
#将 tomcat 运行文件夹的所有者更改为 tomcatadmin
```

以 tomcatadmin 用户身份重新测试 tomcat 启动脚本的执行

```
[root@tomcat /]# netstat -tnl
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address Foreign Address State
tcp 0 0 0.0.0.0:3306 0.0.0.0:* LISTEN
```

```
tcp 0 0 :::80 :::* LISTEN
tcp 0 0 :::22 :::* LISTEN

#查看当前的系统运行状态
[root@tomcat /]# /etc/rc.d/init.d/Tomcat5.sh start
#启动 tomcat 服务

[root@tomcat /]# netstat -tnl
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address Foreign Address State
tcp 0 0 0.0.0.0:3306 0.0.0.0:* LISTEN
tcp 0 0 :::8009 :::* LISTEN
tcp 0 0 :::8080 :::* LISTEN
tcp 0 0 :::80 :::* LISTEN
tcp 0 0 :::22 :::* LISTEN
#查看 tomcat 服务运行状态

[root@tomcat /]# /etc/rc.d/init.d/Tomcat5.sh stop
#停止 tomcat 服务

[root@tomcat /]# netstat -tnl
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address Foreign Address State
tcp 0 0 0.0.0.0:3306 0.0.0.0:* LISTEN
tcp 0 0 :::80 :::* LISTEN
tcp 0 0 :::22 :::* LISTEN
#查看 tomcat 服务是否已经停止
```

将 tomcat 启动脚本注册为系统服务

```
[root@tomcat /]# vi /etc/rc.d/init.d/Tomcat5.sh
#!/bin/sh

#chkconfig: 2345 85 15
#description: Tomcat startup

#在此文件的开始位置添加上述信息，将此文件添加为系统服务
[root@tomcat /]# chkconfig --level 345 Tomcat5.sh on

#将此文件注册为系统服务
```

如果觉得此方式比较的繁琐，也可以将 tomcat/bin 文件夹下的 startup.sh 添加到/etc/rc.d/rc.local 文件中实现 tomcat 的开机启动。

作者简介：CU 网友 kns1024wh，目前从事 Linux 群集方面的具体工作，之前做过多年的 IT 技术支持、MCT 讲师、及 REDFLAG 的技术合作，技术专长群集、unix 主机、AD 部署等，您可以通过电子邮件 lvsheat@qq.com 或者 Chinaunix 社区与他取得联系。

非 LVM 环境中根分区的调整

ChinaUnix 网友：Jerrywjl

大概是因为 Linux 系统相比于微软等系统具有超强的灵活性和透明度的关系，长久以来从来不乏富于想象力的兄弟姐妹总是不断摩拳擦掌地准备将这种在灵活和透明上建立起来的可调控性发挥到极致。

正因为这样，经过不懈的铤而走险才有像 LVM 缩根、单盘分身 Raid 1 等极度刺激和过瘾的方案不断从本人手中孕育出来。当然测试和总结这些方案的初衷基本上都是来自于很多企业和生产环境中的客户顾头不顾腚的前期部署。我们完全有理由相信，并且也时刻做好准备，在今后肯定会有更多神来的想法让我等游走于这种生死之间。所以今天的非 LVM 环境的分区调整又让我体验了一把刺激。废话不多说了，具体情况大概是这样：

在某生产环境中有一台 Red Hat Enterprise Linux 4 的服务器，只安装了一块 160GB 的磁盘，所有的文件系统都部署在非 LVM 的普通分区环境中。但在当年部署该系统的时候估计那哥们的脑袋让驴踢得不轻，整了一个/boot 分区 100MB，/分区为 120GB，数据分区为 40GB 的结构。结果显而易见，该服务器没跑多长时间，存储数据的文件系统即开始告急，但客户又不想将数据放在根分区下，估计用过 Windows 上类似分区魔术师之类的软件，所以自然就又打起了调整根文件系统的主意。要实现的目的如图：

调整前：

| /boot 100MB | ----------/ 120GB---------- | -------/data 40GB------- |

调整后：

| /boot 100MB | --------/ 40GB-------- | ---------- /data 120GB ---------- |

这个对比图相信大家都看得明白，无损数据是基本要求。

这的确是个很有意思和很刺激的需求。其实要说做到也并不能难。尽管没有 LVM 的环境，这种做法不单适用于分区的缩小，而且也可适用于非 LVM 环境中的分区的扩大，而且事实上缩小的操作要比扩大的操作难度更大。但事实上个人认为关键的问题不在于扩大和缩小，而就在于一个字——“算”。我没有办法找到和用户一样的环境，除非有人愿意给我捐一块盘，所以只有找一个虚拟机来模拟测试环境。

这是一个用虚拟机构建一个 RHEL4.7 的系统，安装在一个 4GB 的磁盘上，该系统的分区结构如下：

```
[root@localhost ~]# fdisk -l

Disk /dev/sda: 4294 MB, 4294967296 bytes

255 heads, 63 sectors/track, 522 cylinders

Units = cylinders of 16065 * 512 = 8225280 bytes


   Device Boot      Start         End      Blocks   Id  System

/dev/sda1   *           1          13      104391   83  Linux

/dev/sda2              14         274     2096482+  83  Linux
```

```
/dev/sda3          275       522     1992060   83  Linux
```

而 df 文件系统：

```
[root@localhost ~]# df -TH
Filesystem    Type    Size   Used  Avail Use% Mounted on
/dev/sda2     ext3    2.2G   802M   1.3G  40% /
/dev/sda1     ext3    104M    13M    86M  13% /boot
none          tmpfs   131M     0   131M   0% /dev/shm
/dev/sda3     ext3    2.1G    44M   1.9G   3% /data1
[root@localhost ~]# df
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/sda2              2063536    782996   1175716  40% /
/dev/sda1               101086     12222     83645  13% /boot
none                    127332         0    127332   0% /dev/shm
/dev/sda3              1960716     42220   1818896   3% /data1
```

我的目标是最大化地将根分区缩小。根据这里显示文件系统和块设备的大小，我的目标是将原本存在于/dev/sda2 这个 2GB 磁盘分区上的根文件系统和/dev/sda2 分区本身缩小至少 1GB，因为系统当初是采用 RHEL4 的最小化安装，所以应该只有 800GB 左右；而同时将压缩出来的空间提供给/dev/sda3。

现在开始进行操作。由于要对根进行操作，所以无论如何也得进入 rescue 先。操作之前先计算一下，如果根缩小 1GB 的话，剩下的空间是否满足文件系统的要求：

1GB = 1024 x 1024KB = 1048576KB；

2063536KB – 1048576KB = 1014960KB；

这证明即便缩小根文件系统 1GB，也能为当前的系统根提供足够空间（782996KB）。所以就以此为目标开干！先进入到 rescue 模式下，在提示是否挂载系统根的时候选择“skip”，以备后续手动挂载文件系统。创建目录/mnt/sysimage 备用。现在开始对根分区进行调整。

首先检查该文件系统：# e2fsck –f /dev/sda2

调整 journal：# tune2fs -0 has_journal /dev/sda2

然后缩小该文件系统到所需大小：# resize2fs /dev/sda2 1014960k

```
When finished please exit from the shell and your system will reboot.

-/bin/sh-3.00# mkdir /mnt/sysimage
-/bin/sh-3.00# e2fsck -f /dev/sda2
e2fsck 1.35 (28-Feb-2004)
/: recovering journal
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information

/: ***** FILE SYSTEM WAS MODIFIED *****
/: 30198/262144 files (0.5% non-contiguous), 203985/524120 blocks
-/bin/sh-3.00# tune2fs -O has_journal /dev/sda2
tune2fs 1.35 (28-Feb-2004)
-/bin/sh-3.00# resize2fs /dev/sda2 1014960k
resize2fs 1.35 (28-Feb-2004)
Resizing the filesystem on /dev/sda2 to 253740 (4k) blocks.
The filesystem on /dev/sda2 is now 253740 blocks long.

-/bin/sh-3.00# _
```

在这个操作成功之后，需要挂载/dev/sda2 和原来的/dev/sda3 以测试文件系统通过刚才的折腾是否 OK。

```
-/bin/sh-3.00# mount /dev/sda2 /mnt/sysimage/
-/bin/sh-3.00# mount /dev/sda3 /mnt/sysimage/data1/
-/bin/sh-3.00# df
Filesystem           1K-blocks      Used Available Use% Mounted on
rootfs                    6120      4708      1062  82% /
/dev/root.old             6120      4708      1062  82% /
/tmp/loop0              179160    179160         0 100% /mnt/runtime
/dev/sda2               998472    782488    165236  83% /mnt/sysimage
/dev/sda3              1960716     42220   1818896   3% /mnt/sysimage/data1
-/bin/sh-3.00# fdisk -l /dev/sda

Disk /dev/sda: 4294 MB, 4294967296 bytes
255 heads, 63 sectors/track, 522 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          13      104391   83  Linux
/dev/sda2              14         274     2096482+  83  Linux
/dev/sda3             275         522     1992060   83  Linux
-/bin/sh-3.00#
```

同时查看 fdisk –l 的信息，准备将根分区所在的块设备 sda 缩小到合适大小。

这个地方是最考人的，由于 fdisk 严格来说是以柱面大小为单位计算容量的，所以我们要将已经缩小之后的文件系统换算成柱面数量，再决定当前/dev/sda2 这个文件系统到底需要多少柱面。比如这里根据图示缩小之后的文件系统只有 998472KB，按照 df 命令 1KB 为单位计算的。而通过 fdsik

-l 显示的结果，每个柱面为 8225280 字节，所以现在开始换算：

998472 x 1024 / 8225280 = 124.30401 柱面

也就是说，精确角度来说需要为/dev/sda2 分配 124.30401 个柱面，而这里为了保证分配的柱面数足够而应该分配整数的柱面数，即 125 个；而且同时不要忘了。从 fdisk -l 的结果看来，sda2 原本从 14 柱面开始，但并不意味着其起始位置就可丁可卯地在第 14 柱面开头，因为一个柱面有接近 8MB，换句话说，如果该分区开始的位置在 14 柱面的靠后部分则还需要分配额外的空间。考虑到这种情况可能出现，所以需要再增加一个柱面，即 126 个柱面。即新的/dev/sda2 起始和终止位置应该是 14 柱面到 140 柱面。

因此在大致的考虑和计算之后可以进行分区的调整。方法是运行 fdisk /dev/sda，将/dev/sda2 删除，但注意千万不要保存分区表，然后接着直接建立新的/dev/sda2 分区。将起始和终止柱面设置为 14 和 140 柱面，然后保存分区表。

```
~/bin/sh-3.00# fdisk /dev/sda

Command (m for help): d
Partition number (1-4): 2

Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 2
First cylinder (14-522, default 14):
Using default value 14
Last cylinder or +size or +sizeM or +sizeK (14-274, default 274): 140

Command (m for help): p

Disk /dev/sda: 4294 MB, 4294967296 bytes
255 heads, 63 sectors/track, 522 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          13      104391   83  Linux
/dev/sda2              14         140     1020127+  83  Linux
/dev/sda3             275         522     1992060   83  Linux

Command (m for help): w
```

然后再次挂载文件系统并运行 resize2fs 自动调整文件系统到合适大小。

这个操作完成之后，首先可以确认的一点是分区和文件系统都调整成功了！而且从刚才最后一次 df 命令显示微调结果看来，文件系统扩大了 5MB 左右。证明刚才增加的 126 柱面的算法还是正确的。

现在另外一个关键的问题就是如何压缩出来的磁盘空间来扩展 sda3。事实上如果可用空间在/dev/sda3 文件系统的后面，那么 resize2fs 是可以的，因为这种情况下原来文件系统的 superblock 位置是不变的。但是现在这个情况正好相反，可用空间在/dev/sda3 的前面部分，所以如果按照刚才的方法来强行扩大分区将造成文件系统无法被读取。

```
/dev/root.old              6120      4707      1063  82% /
/tmp/loop0               179160    179160         0 100% /mnt/runtime
/dev/sda2                998480    782460    165272  83% /mnt/sysimage
-/bin/sh-3.00# umount /dev/sda2
-/bin/sh-3.00# e2fsck -f /dev/sda2
e2fsck 1.35 (28-Feb-2004)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/: 30175/131072 files (0.6% non-contiguous), 199737/253742 blocks
-/bin/sh-3.00# resize2fs /dev/sda2
resize2fs 1.35 (28-Feb-2004)
Resizing the filesystem on /dev/sda2 to 255031 (4k) blocks.
The filesystem on /dev/sda2 is now 255031 blocks long.

-/bin/sh-3.00# df
Filesystem           1K-blocks      Used Available Use% Mounted on
rootfs                    6120      4707      1063  82% /
/dev/root.old             6120      4707      1063  82% /
/tmp/loop0              179160    179160         0 100% /mnt/runtime
-/bin/sh-3.00# mount /dev/sda2 /mnt/sysimage/
-/bin/sh-3.00# df
Filesystem           1K-blocks      Used Available Use% Mounted on
rootfs                    6120      4707      1063  82% /
/dev/root.old             6120      4707      1063  82% /
/tmp/loop0              179160    179160         0 100% /mnt/runtime
/dev/sda2              1003636    782460    180372  82% /mnt/sysimage
-/bin/sh-3.00#
```

对于这个问题我们目前没有更好的方法，只有将原来的/dev/sda3 文件系统先备份出来，然后在这个环境中将/dev/sda2 删除。再通过 fdisk 重新建立并 mkfs 创建文件系统。

为了测试需要，我随便拷贝一点数据上去，然后备份该文件系统的内容到/mnt/sysimage 目录下。

```
-/bin/sh-3.00# df
Filesystem           1K-blocks      Used Available Use% Mounted on
rootfs                    6120      4707      1063  82% /
/dev/root.old             6120      4707      1063  82% /
/tmp/loop0              179160    179160         0 100% /mnt/runtime
/dev/sda2               998480    782460    165272  83% /mnt/sysimage
-/bin/sh-3.00# mount /dev/sda3 /mnt/sysimage/data1
-/bin/sh-3.00# df
Filesystem           1K-blocks      Used Available Use% Mounted on
rootfs                    6120      4707      1063  82% /
/dev/root.old             6120      4707      1063  82% /
/tmp/loop0              179160    179160         0 100% /mnt/runtime
/dev/sda2              1003636    782460    180372  82% /mnt/sysimage
/dev/sda3              1960716     42220   1818896   3% /mnt/sysimage/data1
-/bin/sh-3.00# ls /mnt/sysimage/data1/
anaconda-ks.cfg             install.log.syslog         System.map-2.6.9-78.EL
config-2.6.9-78.EL          lost+found                 System.map-2.6.9-78.ELsmp
config-2.6.9-78.ELsmp       message                    vmlinuz-2.6.9-78.EL
initrd-2.6.9-78.EL.img      message.ja                 vmlinuz-2.6.9-78.ELsmp
initrd-2.6.9-78.ELsmp.img   symvers-2.6.9-78.EL.gz
install.log                 symvers-2.6.9-78.ELsmp.gz
-/bin/sh-3.00# dump -0f /mnt/sysimage/sda3 /dev/sda3
  DUMP: Can't open /etc/fstab for dump table information: No such file or direct
ory
  DUMP: WARNING: no file `/etc/dumpdates'
  DUMP: Date of this level 0 dump: Thu Aug 27 14:57:54 2009
  DUMP: Dumping /dev/sda3 (/mnt/sysimage/data1) to /mnt/sysimage/sda3
  DUMP: Label: none
  DUMP: Writing 10 Kilobyte records
  DUMP: mapping (Pass I) [regular files]
```

这种方法可完成整个文件系统层面的备份，而且备份的数据和实际数据大小一致。当然也可以

用 tar 等工具，反正只要完成备份即可。最后删除原来的/dev/sda3 利用现在的起始柱面和默认的终止柱面重建 sda3，并格式化。然后将备份集还原，到此一切操作即成功！

```
-/bin/sh-3.00# mount /dev/sda3 /mnt/sysimage/data1/
-/bin/sh-3.00# pwd
/mnt/sysimage/data1
-/bin/sh-3.00# restore -rf /mnt/sysimage/sda3
restore: ./lost+found: File exists
-/bin/sh-3.00# pwd
/mnt/sysimage/data1
-/bin/sh-3.00# ls
anaconda-ks.cfg             install.log.syslog      symvers-2.6.9-78.ELsmp.gz
config-2.6.9-78.EL          lost+found              System.map-2.6.9-78.EL
config-2.6.9-78.ELsmp       message                 System.map-2.6.9-78.ELsmp
initrd-2.6.9-78.EL.img      message.ja              vmlinuz-2.6.9-78.EL
initrd-2.6.9-78.ELsmp.img   restoresymtable        vmlinuz-2.6.9-78.ELsmp
install.log                 symvers-2.6.9-78.EL.gz
-/bin/sh-3.00# df
Filesystem           1K-blocks      Used Available Use% Mounted on
rootfs                    6120      4708      1062  82% /
/dev/root.old             6120      4708      1062  82% /
/tmp/loop0              179160    179160         0 100% /mnt/runtime
/dev/sda2              1003636    789028    173804  82% /mnt/sysimage
/dev/sda3              3020172     44020   2822732   2% /mnt/sysimage/data1
-/bin/sh-3.00# _
```

最后重启整个系统进入正常模式，一切正常，调整顺利完成！

纵观整个测试过程，美中不足的是在这种情况下没法对/dev/sda3 进行调整，当然可用空间再/dev/sda3 文件系统的后面部分又是另外一回事。

其实这种情况也可以使用一些第三方工具如 qtparted 等试试。应该也可以有不错的效果。在这个试验中，主要是我们设计的环境比较险恶。不过这完全可以回应一些对调整非 LVM 上 ext 文件系统和块设备大小有需求的哥们。

不过不管怎么说，在 LVM 上操作时标准和科学的做法，毕竟在那个环境中软件在帮你进行最关键的计算；而这种冒险的方法，除非你能很好把握，否则最好不要轻易尝试。

在 Linux 下通过 PPP 上 WCDMA

ChinaUnix 网友：yaofei

折腾了一番通过几种 USB 上网卡在 Linux 下上 WCDMA，在此记录一下：

通用配置文件：

/etc/ppp/peers/wcdma :

/dev/ttyACM0

460800

connect '/etc/ppp/chat-wcdma'

noauth

usepeerdns

noipdefault

defaultroute

注意，其中的/dev/ttyACM0 是串口设备，后面描述。

/etc/ppp/chat-wcdma

```
#!/bin/sh
#
# This is part 2 of the ppp-on script. It will perform the connection
# protocol for the desired connection.
#
exec /usr/sbin/chat -v \
ECHO ON \
ABORT 'BUSY' \
ABORT 'NO ANSWER' \
ABORT 'ERROR' \
TIMEOUT 20 \
'' 'AT' \
OK AT+CFUN=6 \
      OK          'AT+CGDCONT=1,"IP","3gnet",,0,0' \
OK 'ATDT*99#' \
CONNECT
```

其中的 3gnet 就是联通 WCDMA 上网需要的 APN 啦。 CFUN=6 是让索爱 MD-400 只工作在 WCDMA 网络下。相应地 CFUN=5 就只工作再 GSM/GPRS/EDGE 下， CFUN=1 就是全自动选择。

拨号连接时，用

pppd call wcdma

即可。你可以自己设置 IP 伪装 NAT 等等事宜。

以上是用索尼爱立信 MD-400 上网卡设置的，如果你用的是华为 E1750, 那么设备名就换成/dev/ttyUSB0 即可。

无论是索爱 MD400 还是华为 E1750, 都需要小工具 usb_modeswitch 1.0.2 切换其倒霉的 USB 工作状态，详细的介绍和下载可以去这里看看： http://www.draisberghof.de/usb_modeswitch/

如果是电信的天翼 CDMA2000 EV-DO 3G, 测试了一下华为 EC1260，基本区别不大，注意以下几点：

1， 2.6.19 之后的核心有专门对付 EC1260 的代码, 插进去直接就令其工作在 Modem 模式，你会看

到有/dev/ttyUSB0 设备，可以不需要 usb_modeswitch 的帮助。低版本核心可能还离不开。

2， EVDO 没有 APN 的概念，上面的/etc/ppp/chat-wcdma 脚本改成这样即可：

```
[root@pxi4g ppp]# cat chat-evdo
#!/bin/sh
#
# This is part 2 of the ppp-on script. It will perform the connection
# protocol for the desired connection.
#
exec /usr/sbin/chat -v \
ECHO ON \
ABORT 'BUSY' \
ABORT 'NO ANSWER' \
ABORT 'ERROR' \
TIMEOUT 20 \
'' 'AT' \
OK 'ATDT#777' \
CONNECT
```

也就是说，直接拨特殊号码#777 即可。 peers/wcdma 文件可以不改。

CentOS5.2 服务器上使用 KVM 进行虚拟化应用

ChinaUnix 网友： tinybiz（译）

原文：http://www.howtoforge.com/virtualization-with-kvm-on-a-centos-5.2-server

这篇教程将会为你详细描述怎样在一台 CentOS5.2 服务器上安装和使用 KVM，来创建和运行虚拟机，我不仅教大家怎样创建 image-based 虚拟机，同样也教大家创建一台 LVM 的虚拟机.KVM 是 Kernel-based Virtual Machine 的缩写，使用的是硬件虚拟化的技术，换言之，你的 CPU 需要支持硬件虚拟化，例如 Intel VT 或者 AMD-V 技术。

我不保证你参考这篇教程完全能使你正常工作!

1 前言

我使用的一台主机名为 server1.example.com 和 ip 地址是 192.168.0.100 的 CentOS5.2 服务器作为我的 KVM 主机。

在这里我也需要一个安装有 virt-manager 的客户端系统，可以使我们能够连接到虚拟机的图形终端。我在这里使用的是 Ubuntu 8.10 桌面版。

2 安装 KVM

CentOS 5.2 KVM 主机：

运行

system-config-securitylevel

并且开启 SELinux（如果你的 SELinux 被禁用，virt-install 将不会正常工作）。

然后检查 CPU 是否支持硬件虚拟化-运行命令

egrep '(vmx|svm)' --color=always /proc/cpuinfo

应该会显示如下信息：

[root@server1 ~]# egrep '(vmx|svm)' --color=always /proc/cpuinfo
flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 c lflush mmx fxsr sse sse2 ht syscall
nx mmxext fxsr_opt rdtscp lm 3dnowext 3dnow pni cx16 lahf_lm cmp_legacy svm extapic cr8_legacy misalignsse
flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 c lflush mmx fxsr sse sse2 ht syscall
nx mmxext fxsr_opt rdtscp lm 3dnowext 3dnow pni cx16 lahf_lm cmp_legacy svm extapic cr8_legacy misalignsse
[root@server1 ~]#

如果什么也没有显示的话，就说明你的处理器不支持硬件虚拟化技术，下面的内容就不适合你了。

下面我们为软件包导入 GPG key

rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY*

然后安装 KVM 和 virtinst（一个创建虚拟机的工具），我们运行

yum install kvm kmod-kvm qemu libvirt python-virtinst

然后重新启动系统：

reboot

启动过后，KVM 的内核模块将会被加载：

lsmod | grep kvm

[root@server1 ~]# lsmod | grep kvm
kvm_amd 50452 0
kvm 109264 1 kvm_amd
[root@server1 ~]#

（这个输出结果是一台拥有 AMD-V 处理器的系统，如果你的电脑使用的是 Intel VT 的 CPU，将会显示类似 *kvm_intel 字符*）

使用下列命令检查 KVM 是否成功安装

virsh -c qemu:///system list

将会显示如下结果：

[root@server1 ~]# virsh -c qemu:///system list
Id Name State

[root@server1 ~]#

如果在这里显示的是一个错误的信息，说明有些东西出现了问题。

下面我们需要在我们的服务器上设置一个网桥，就可以使我们的虚拟机从其他主机中读取数据。

要做到这一点，我们安装 bridge-utils 工具。。。。。。

yum install bridge-utils

... 并且配置一个网桥从/etc/sysconfig/network-scripts/ifcfg-eth0 参考 BOOTPROTO, BROADCAST, IPADDR, NETMASK 和 NETWORK 等值来创建文件/etc/sysconfig/network-scripts/ifcfg-br0：

vi /etc/sysconfig/network-scripts/ifcfg-br0

```
DEVICE=br0
TYPE=Bridge
BOOTPROTO=static
BROADCAST=192.168.0.255
IPADDR=192.168.0.100
NETMASK=255.255.255.0
NETWORK=192.168.0.0
ONBOOT=yes
```

修改*/etc/sysconfig/network-scripts/ifcfg-eth0* as follows (取消注释 *BOOTPROTO*, *BROADCAST*, *IPADDR*, *NETMASK*, 和 *NETWORK* 最后添加 *BRIDGE=br0*):

vi /etc/sysconfig/network-scripts/ifcfg-eth0

```
# Realtek Semiconductor Co., Ltd. RTL-8139/8139C/8139C+
DEVICE=eth0
#BOOTPROTO=static
#BROADCAST=192.168.0.255
HWADDR=00:10:A7:05:AF:EB
#IPADDR=192.168.0.100
#NETMASK=255.255.255.0
```

```
#NETWORK=192.168.0.0
ONBOOT=yes
BRIDGE=br0
```

重新启动网络...

/etc/init.d/network restart

... 运行

ifconfig

现在应该显示网桥(*br0*):

```
[root@server1 ~]# ifconfig
br0       Link encap:Ethernet  HWaddr 00:10:A7:05:AF:EB
          inet addr:192.168.0.100  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::210:a7ff:fe05:afeb/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:17 errors:0 dropped:0 overruns:0 frame:0
          TX packets:53 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1160 (1.1 KiB)  TX bytes:14875 (14.5 KiB)

eth0      Link encap:Ethernet  HWaddr 00:10:A7:05:AF:EB
          inet6 addr: fe80::210:a7ff:fe05:afeb/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:13662 errors:7 dropped:160 overruns:4 frame:0
          TX packets:11646 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:15144608 (14.4 MiB)  TX bytes:1379942 (1.3 MiB)
          Interrupt:74 Base address:0xcc00

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:38 errors:0 dropped:0 overruns:0 frame:0
          TX packets:38 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:4308 (4.2 KiB)  TX bytes:4308 (4.2 KiB)

virbr0    Link encap:Ethernet  HWaddr 00:00:00:00:00:00
          inet addr:192.168.122.1  Bcast:192.168.122.255  Mask:255.255.255.0
          inet6 addr: fe80::200:ff:fe00:0/64 Scope:Link
```

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:35 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:0 (0.0 b) TX bytes:9987 (9.7 KiB)

[root@server1 ~]#

3 在你的 Ubuntu 8.10 Desktop 上安装 virt-viewer 或者 virt-manager

Ubuntu 8.10 Desktop:

在这里我们需要一个连接我们客户机图形终端的一个工具-我们可以使用 virt-manager 来做到这一点（请参考 KVM Guest Management With Virt-Manager On Ubuntu 8.10）。我在这里假设我使用的是 Ubuntu 8.10 Desktop。

运行

sudo aptitude install virt-manager

来安装 virt-manager.

（如果你是用的是 Fedora 10 desktop，你可以使用下列命令安装 virt-manager：

先成为 root。。。

su

... 然后运行

yum install virt-manager

）

4 创建 Debian Lenny 客户机（Image-Based）

CentOs 5.2 KVM 主机:

现在让我们回到我们的 CentOS 5.2 KVM 主机：

先参考一下

man virt-install

来学习如何使用 virt-install.

我们使用 bridging 模式创建一个名字为 vm10，512 内存，2 个虚拟 CPU，磁盘镜像为 ~/vm10.gcow2(有 12G 大小)Debian Lenny 客户机，并且插入 Debian Lenny Netinstall CD 到光驱，运行

virt-install --connect qemu:///system -n vm10 -r 512 --vcpus=2 -f ~/vm10.qcow2 -s 12 -c /dev/cdrom --vnc --noautoconsole --os-type linux --os-variant generic26 --accelerate

--network=bridge:br0 --hvm

（virt-install 的 man 文件显示了--os-type 和--os-variant 合法的值。CentOS 5。2 中附带的 virt-install 的版本不能识别 Debian Lenny，所以我们使用 generic26 来代替--os-variant.）

当然，你也可以常见一个 Debian Lenny Netinstall CD 的 ISO 镜像。。。

dd if=/dev/cdrom of=~/debian-500-amd64-netinst.iso

。。。通过 virt-install 命令使用 ISO 镜像：

virt-install --connect qemu:///system -n vm10 -r 512 --vcpus=2 -f ~/vm10.qcow2 -s 12 -c ~/debian-500-amd64-netinst.iso --vnc --noautoconsole --os-type linux --os-variant generic26 --accelerate --network=bridge:br0 --hvm

输出结果如下所示：

[root@server1 ~]# virt-install --connect qemu:///system -n vm10 -r 512 --vcpus=2 -f ~/vm10.qcow2 -s 12 -c ~/debian-500-amd64-netinst.iso --vnc --noautoconsole --os-type linux --os-variant generic26 --accelerate --network=bridge:br0 --hvm

Starting install...
Creating storage file... 100% |=========================| 12 GB 00:00
Creating domain... 0 B 00:00
Domain installation still in progress. You can reconnect to
the console to complete the installation process.
[root@server1 ~]#

5 连接到客户机

Ubuntu 8.10 Desktop:

KVM 客户机将会从 Debian Lenny Netinstall CD 启动并且启动 Debian installer-这就是我们必须连接到客户机图形终端的原因。你可以通过在 Ubuntu 8.10 desktop 使用 virt-manager 来做到这一点。

运行

sudo virt-manager

在 Ubuntu desktop 上启动 virt-manager.

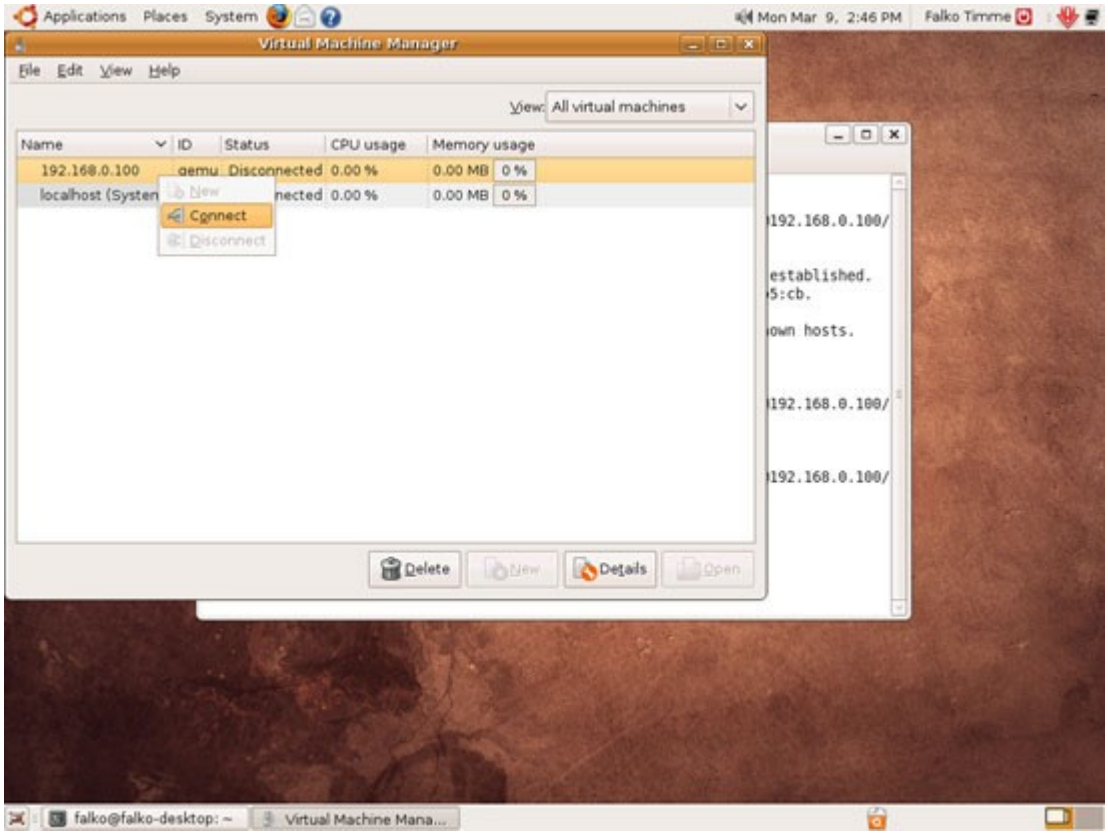
（如果你是 Fedora 10 desktop，运行：

su

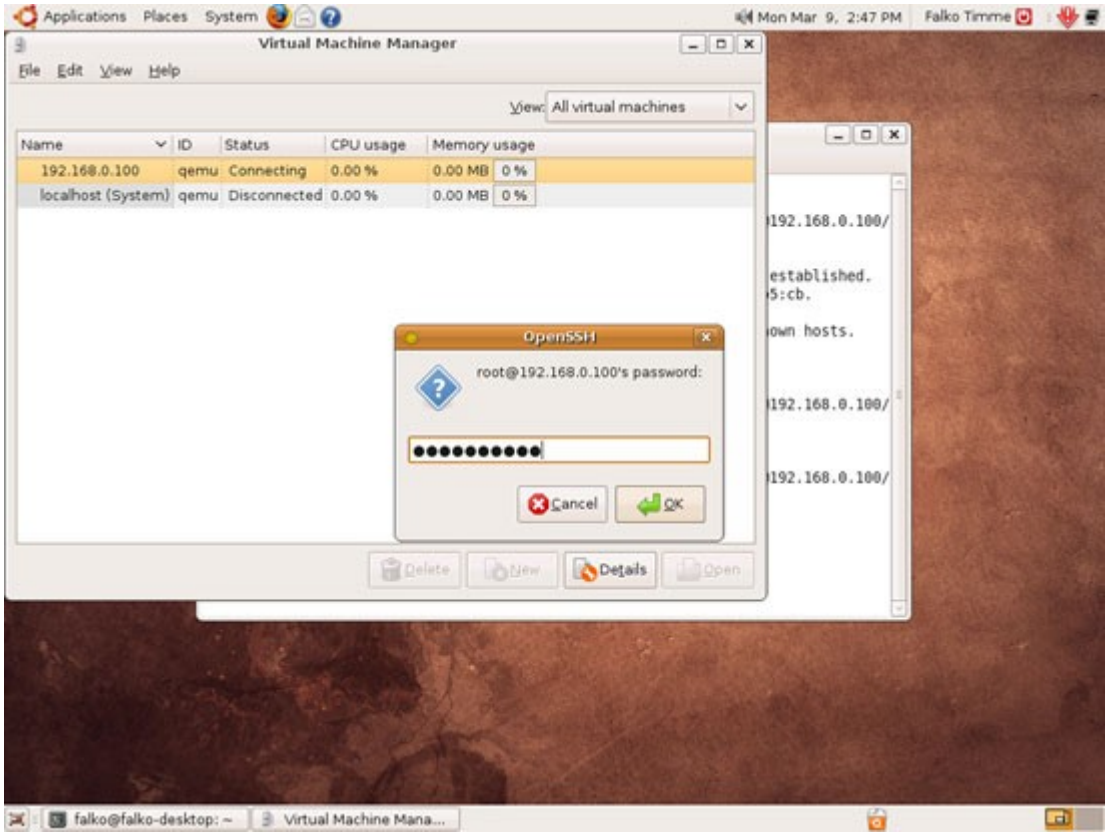
virt-manager

）

在 virt-manager，连接到 KVM 主机：

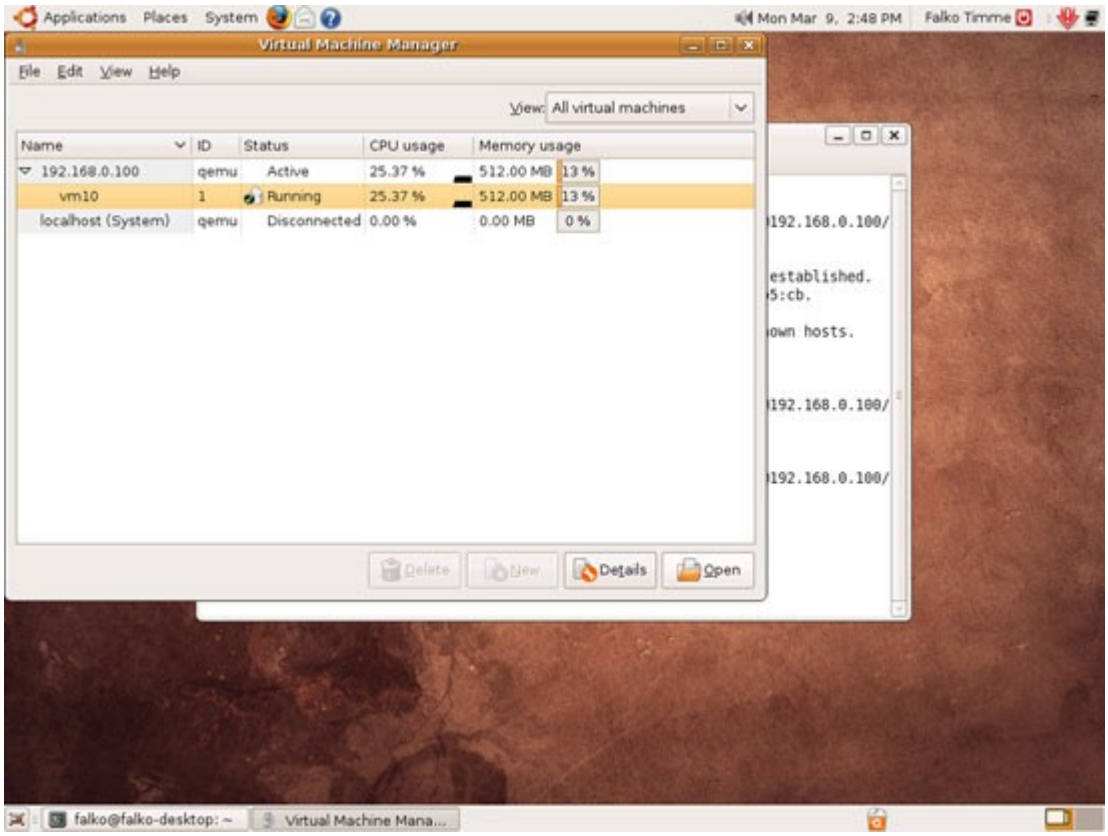


输入 KVM 主机的 root 密码：

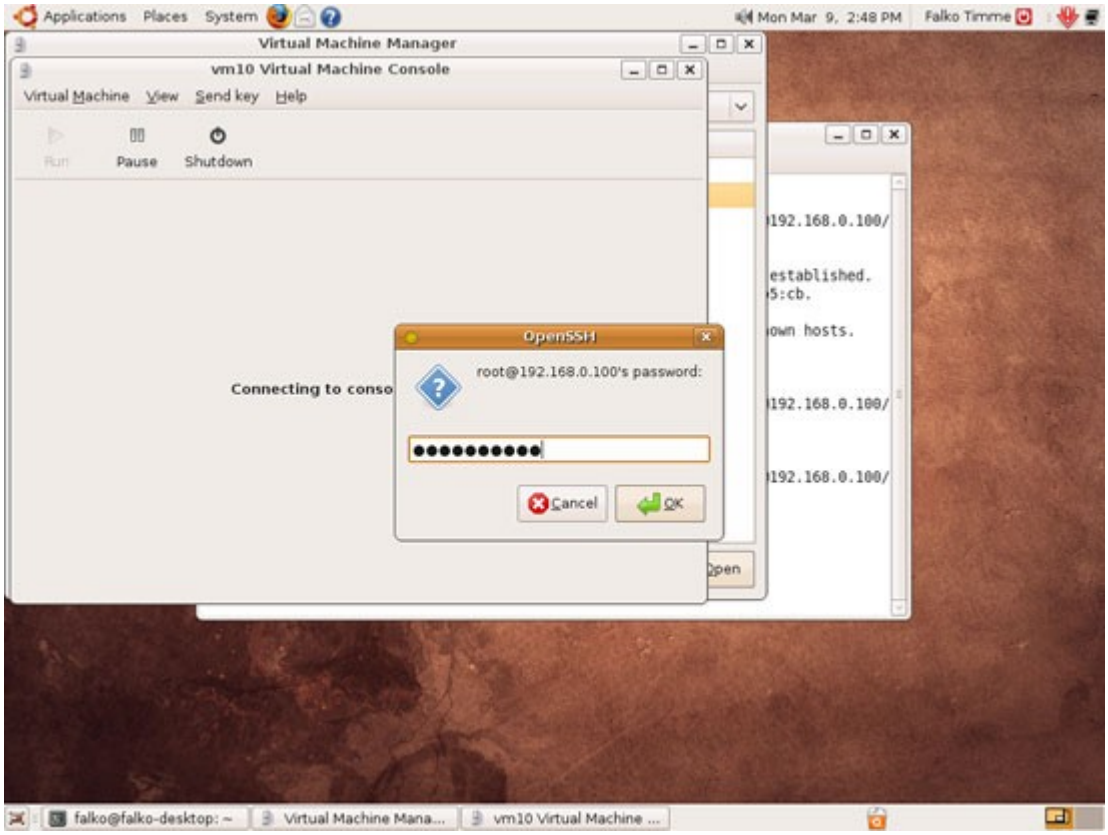


输入完以后，你就会看到 vm10 这个主机正在运行了，标注客户机并且点击 *Open* 按钮来打开客户

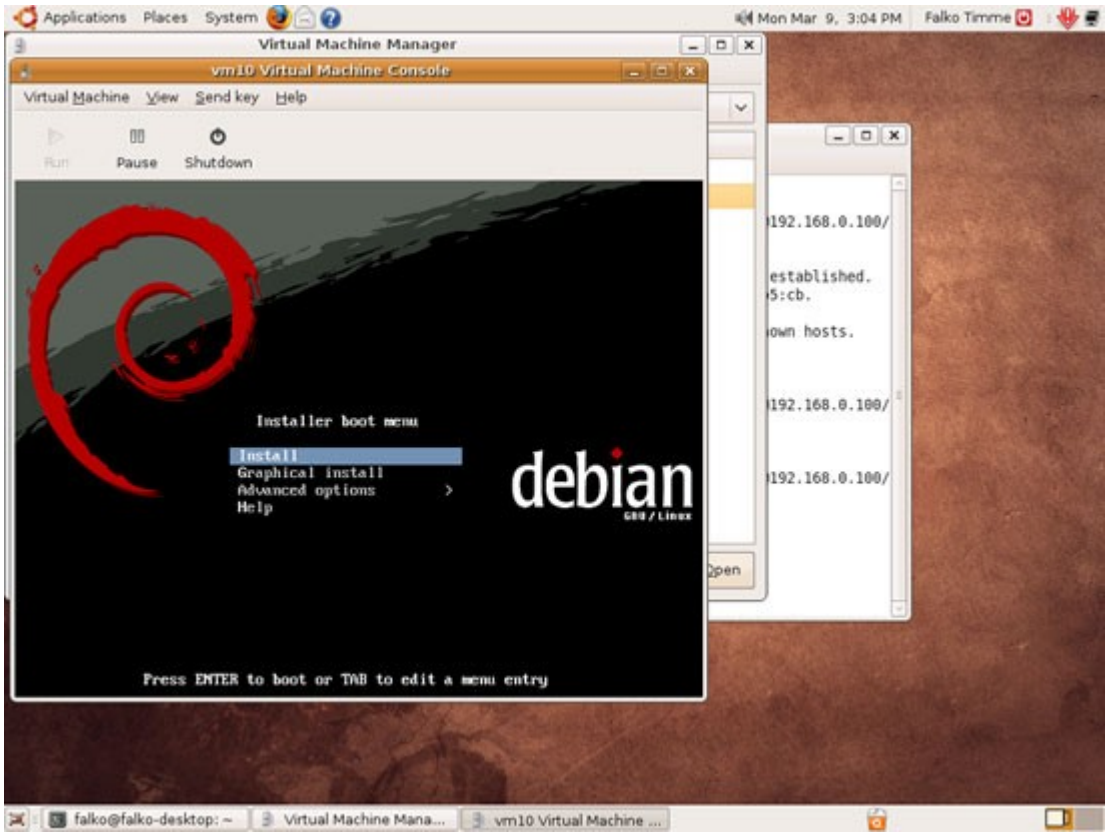
机的图形终端：



再次输入 KVM 主机的 root 密码：



你现在就可以连接到客户机的图形终端了。并且能看到 Debian installer:



现在你既可以像在物理系统中安装 Debian 一样正常安装 Debian 了。请注意在安装结束后，Debian 客户机需要重启。客户机将会停掉，所以你需要再次重启它。可以使用 virt-manager 也可以使用我们的 CentOS 5.2 主机上的命令：

CentOS 5.2 KVM 主机:

virsh --connect qemu:///system

start vm10

quit

然后，你就可以使用 virt-manager 连接到客户机并且配置它。如果你在客户机中安装了 OpenSSH（openssh-server 包），你可以通过 SSH 客户端连接它（类似 PuTTY）。

6 管理 KVM 客户机

CentOS 5.2 KVM 主机:

可以通过 *virsh* 命令管理 KVM 客户机，"virtual shell"。连接到 virtual shell，运行

virsh --connect qemu:///system

下面就是 virtual shell 的显示界面：

[root@server1 ~]# virsh --connect qemu:///system
Welcome to virsh, the virtualization interactive terminal.

Type: 'help' for help with commands
'quit' to quit

virsh #

现在你可以在 virtual shell 中输入命令来管理你的客户机，运行：

help

获得更多的命令：

virsh # help
Commands:

help print help
attach-device attach device from an XML file
attach-disk attach disk device
attach-interface attach network interface
autostart autostart a domain
capabilities capabilities
connect (re)connect to hypervisor
console connect to the guest console
create create a domain from an XML file
start start a (previously defined) inactive domain
destroy destroy a domain
detach-device detach device from an XML file
detach-disk detach disk device
detach-interface detach network interface
define define (but don't start) a domain from an XML file
domid convert a domain name or UUID to domain id
domuuid convert a domain name or id to domain UUID
dominfo domain information
domname convert a domain id or UUID to domain name
domstate domain state
domblkstat get device block stats for a domain
domifstat get network interface stats for a domain
dumpxml domain information in XML
freecell NUMA free memory
hostname print the hypervisor hostname
list list domains
migrate migrate domain to another host
net-autostart autostart a network
net-create create a network from an XML file
net-define define (but don't start) a network from an XML file
net-destroy destroy a network

net-dumpxml network information in XML
net-list list networks
net-name convert a network UUID to network name
net-start start a (previously defined) inactive network
net-undefine undefine an inactive network
net-uuid convert a network name to network UUID
nodeinfo node information
quit quit this interactive terminal
reboot reboot a domain
restore restore a domain from a saved state in a file
resume resume a domain
save save a domain state to a file
schedinfo show/set scheduler parameters
dump dump the core of a domain to a file for analysis
shutdown gracefully shutdown a domain
setmem change memory allocation
setmaxmem change maximum memory limit
setvcpus change number of virtual CPUs
suspend suspend a domain
ttyconsole tty console
undefine undefine an inactive domain
uri print the hypervisor canonical URI
vcpuinfo domain vcpu information
vcpupin control domain vcpu affinity
version show version
vncdisplay vnc display

virsh #

list

显示所有正在运行的客户机；

list --all

显示所有客户机，正在运行的和没有运行的：

```
virsh # list --all
 Id Name                 State
----------------------------------
  2 vm10                 running

virsh #
```

如果你修改了一个客户机的 xml 文件（位于*/etc/libvirt/qemu/* 目录），你必须重新定义客户机：

define /etc/libvirt/qemu/vm10.xml

请注意，无论你何时在*/etc/libvirt/qemu/* 中修改了客户机的 XML 文件，你必须重新运行 *define* 命令！

启动和停止客户机，运行：

start vm10

停止一个客户机，运行

shutdown vm10

立即中断一个客户机（类似直接关电源），运行

destroy vm10

挂起一个客户机:

suspend vm10

恢复客户机:

resume vm10

这些都是最重要的命令.

输入

quit

退出 virtual shell.

7 创建一个 LVM-Based 客户机

CentOS 5.2 KVM 主机:

LVM-based 客户机与 image-based 客户机相比较而言有很多优势。LVM-based 不但减轻了硬盘 IO 的负担，而且很方便就可以备份（使用 LVM snapshots）。

使用 LVM-based 客户机，你需要有剩余空间的卷组，并且此卷组没有分配给任何逻辑卷。在这个例子中，我使用的卷组叫做*/dev/VolGroup00* 容量大小大约有 148GB。。。

gdisplay

```
[root@server1 ~]# vgdisplay
  /dev/hda: open failed: No medium found
  --- Volume group ---
  VG Name               VolGroup00
  System ID
  Format                lvm2
  Metadata Areas        1
  Metadata Sequence No  3
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
```

```
  Cur LV                2
  Open LV               2
  Max PV                0
  Cur PV                1
  Act PV                1
  VG Size               148.53 GB
  PE Size               32.00 MB
  Total PE              4753
  Alloc PE / Size       968 / 30.25 GB
  Free  PE / Size       3785 / 118.28 GB
  VG UUID               5faE1k-DkMu-JUEk-K0JV-B9ta-Nyaf-n7tngf

[root@server1 ~]#
```

。。。这里面其中包涵/dev/VolGroup00/LogVol00 的卷组大约
30GB，/dev/VolGroup00/LogVol00 的卷组（大约 1GB）-剩余的空间没有分配，可以让 KVM 客户机使用：

lvdisplay

```
[root@server1 ~]# lvdisplay
  /dev/hda: open failed: No medium found
  --- Logical volume ---
  LV Name                /dev/VolGroup00/LogVol00
  VG Name                VolGroup00
  LV UUID                qzC8v6-cLyi-Pr4g-BjJv-35Xr-cEJM-LBVs7G
  LV Write Access        read/write
  LV Status              available
  # open                 1
  LV Size                29.28 GB
  Current LE             937
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           253:0

  --- Logical volume ---
  LV Name                /dev/VolGroup00/LogVol01
  VG Name                VolGroup00
  LV UUID                xA3e1Z-mEc9-rGT1-WcAu-TjF4-lbf3-6LvFaj
  LV Write Access        read/write
  LV Status              available
  # open                 1
  LV Size                992.00 MB
```

```
  Current LE            31
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           253:1

[root@server1 ~]#
```

现在我创建 vm11 的虚拟机作为 LVM-based 的客户机。我预想 vm11 有 20GB 的磁盘空间，因此我创建一个 20GB 大小的名为/dev/VolGroup00/vm11 的逻辑卷：

lvcreate -L20G -n vm11 VolGroup00

然后，我再次使用 virt-install 来创建客户机：

virt-install --connect qemu:///system -n vm11 -r 512 --vcpus=2 -f /dev/VolGroup00/vm11 -c ~/debian-500-amd64-netinst.iso --vnc --noautoconsole --os-type linux --os-variant generic26 --accelerate --network=bridge:br0 --hvm

请注意我在这里使用的是*-f /dev/VolGroup00/vm11 而非 -f ~/vm11.qcow2*，并且我并不需要-s；来重新定义磁盘空间，因为我已经通过 *vm11* (20GB)定义过一个逻辑卷了。

现在我们就可以按照第五章的流程来安装客户机了。

8 链接

1、KVM: http://kvm.qumranet.com/
2、CentOS: http://www.centos.org/
3、Debian: http://www.debian.org/
4、Ubuntu: http://www.ubuntu.com/

关于 bacula 网络备份软件的安装以及配置

ChinaUnix 网友：wiliiwin

最近在网上看到有一个强大的网络备份软件 bacula,于是就找了一些资料来看,发现网上的资料寥寥无几啊,凭着自己的理解就在本地环境下面进行了测试,最后终于实现了网络的备份以及还原功能,当然此文章也有不足之处,之后会进行其他方面的一些更新。

网络备份工具 bacula 配置

简介: Bacula,被誉为开源软件中最好的备份还原软件,它提供了企业级的客户机/服务器的备份解决方案，能够通过网络来管理文件的备份，恢复和核实工作.既有 windows 版本的,也有 Linux，Unix 的。

关于 bacula 的组建介绍:

Directory:管理所有备份,恢复,验证,和存档事务,定制备份和恢复文件的计划.

Storge:指定进行存储和恢复文件属性和数据的物理备份媒介.

File:安装在被备份机器上的程序,将被 directory 调用时候,它提供

关于自己的操作系统、文件属性、数据等资料.

Console:与 directory 进行通讯.

Catalog:负责维护所有备份文件的索引和数据库.

Monitor:监控 directory、file、storage 的守护进程.

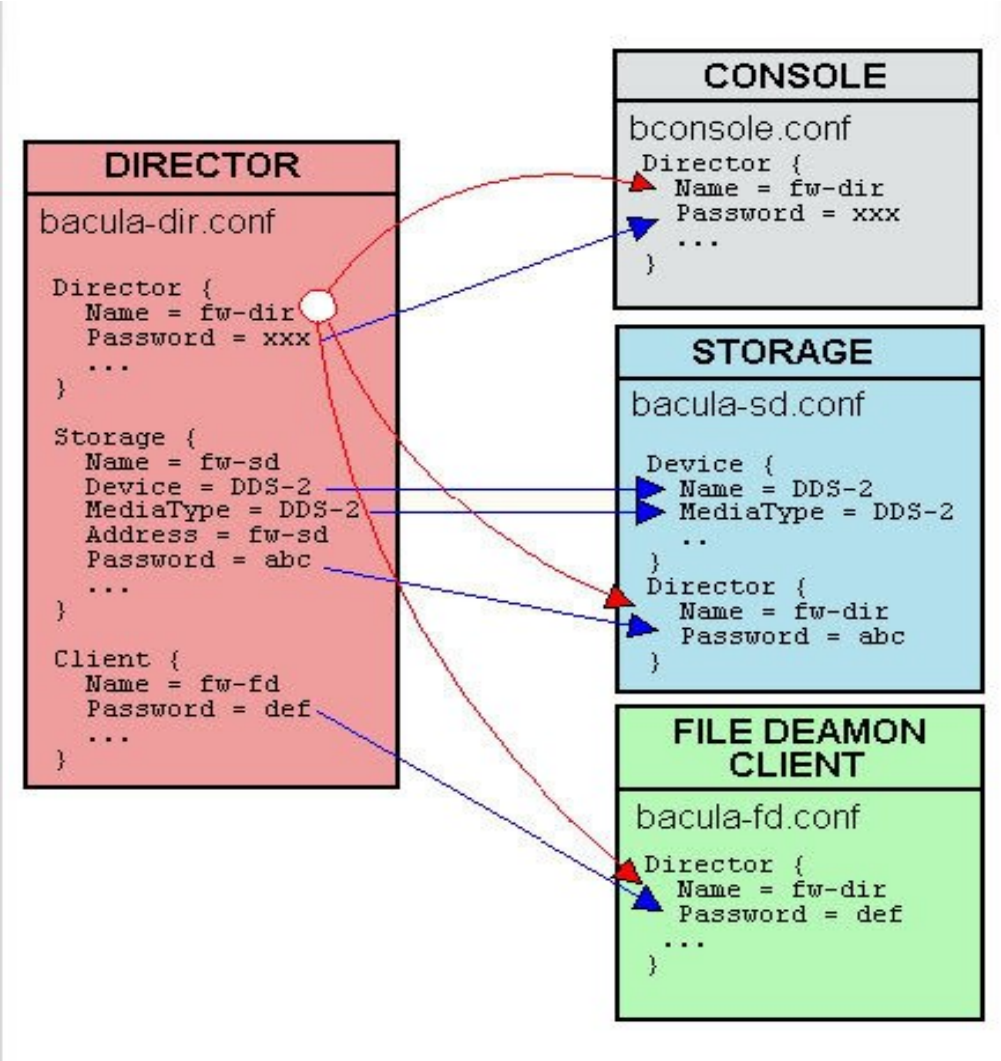
试验环境:

使用 2 台服务器来做这个网络备份的测试环境,环境如下:

10.10.2.226　　作为 directory、storage、catalog.

10.10.2.162　　作为 file.

关于 bacula 的结构图:



一、Bacula 在 server 上的安装以及配置

1.Bacula 在 server 上的安装

bacula 的安装不是很复杂,我这里安装的是 bacula 的 2.4.3 版本的.

#tar zxvf bacula-2.4.3.tar.gz

#./configure --with-mysql

#make&&make intall

默认情况下,bacula 的安装路径为/etc/bacula.

2.创建 mysql 脚本

#./grant_mysql_privileges

#./ create_mysql_database

#./ make_mysql_tables

3.bacula 的配置

在这里说明下,bacula 的主要配置都在 directory 这个配置文件里面的,这个算是一个核心文件了,一定要搞清楚里面配置文件的配置项都代表什么意思,否则很难配置成功.

##

directory 端的配置:

#more bacula-dir.conf

//directory 的全局配置

```
Director {
  Name = linux-0fdr-dir
  DirAddress=10.10.226          //这里是 directory 端的 ip
  DIRport = 9101                //这里是 directory 端的进程号
  QueryFile = "/etc/bacula/query.sql"
  WorkingDirectory = "/var/bacula/working"
  PidDirectory = "/var/run"
  Maximum Concurrent Jobs = 1
  Password = "MJ8SyFywMm+4ntJf2EupggRWIWE6LozmLfIKC8J9nYt7"
  Messages = Daemon
}
```

//directory 的 job 配置

```
Job {
  Name =dbjob        //job 的名字,这个可以随便起,在后面的备份操作的时候可以看到
  Client=dbfd        //这里要和后面的 client 的 name 名字要一样
```

```
  Type=backup        //可用类型有 backup,restore,verify,admin
  FileSet=dbfs       //这个要和后面的 fileset 的 name 名字要一样
  Messages=Daemon    //这个要与后面的 message 的 name 名字要一样
  Pool=dbpool        //这个要与后面的 pool 的 name 名字要一样
  Storage=dbsd       //这个要与后面的 storage 的 name 名字要一样
  Write Bootstrap = "/var/bacula/working/Client1.bsr"
  Level=Full     //这里是备份类型,可用的值是 FULL(完全备份),incremental(增量备
份),differential(差异备份)
  schedule=dbscd     //这里要与后面的 schedule 的 name 名字要一样
}
//directory 的还原 job 配置
job{
  Name=restrory
  Client=dbfd
  Type=Restore
  FileSet=dbfs
  Messages=Daemon
  Pool=dbpool
  Storage=dbsd
  Write Bootstrap = "/var/bacula/working/Client1.bsr"
  Where =/home/bacula/    //还原的路径


}
FileSet {
  Name =dbfs       //这个名字是应用到 job 里面的
  Include{
    Options {
      Compression=GZIP   //对备份文件进行 gzip 的压缩
      signature = MD5    //验证方式为 MD5 码的方式
      Sparse=yes
    }
```

```
    File =/home/mysql_packup   //需要备份的目录
  }
Exclude {              // 排除的，不备份的内容
    File = /proc        //不备份/proce
    File = /tmp
    File = /.journal
    File = /.fsck
  }
}

Client {
  Name = dbfd            //这里要与 FD 配置文件的 name 相同,并且应用在 job 里面的
  Address = 10.10.2.162   //要备份的客户机 IP
  FDPort = 9102
  Catalog = dbcatalog     //记录客户机备份情况的日志名,后面会定义
  Password = "Nd+Yuqe+Kd5wBc1S9uMTEGTNqNIe1cupnfuOdwboej6Z"  //与 FD 的密码一致
File Retention = 30 days  //保存在数据库目录多久轮询一次,这里是 30 天
  Job Retention = 6 months   //job 保持周期,大于 FRetention
  AutoPrune = yes          //当 FR/JR 周期满了是否执行自动删除它们在数据库的目录
  #priority  =1            //如果多个 client 时,定义优先级 1-1000,值越小越优先
}

Storage {
  Name =dbsd      //必须与 SD 配置文件的 name 相同,这个应用到了 job 里面
  Address = 10.10.2.226        //安装 sd 服务的服务器 IP
  SDPort = 9103
  Password = "5jipKA7pekDel6BUjglwD3LkPPBgxcecbLpLfl9puj0c"    //要与 SD 的 password 一
样
  Device =dbdev         //要与 SD 配置文件的 device 的 name 相同
  Media Type = File      //要与 SD 配置文件 device 的 Media Type 相同
}
```

```
Pool {
  Name =dbpool          //这个应用到了 job 里面
  Pool Type=backup
  Maximum Volume Jobs = 1
  Recycle = yes
  AutoPrune = yes
  Volume Retention = 365 days
}

Schedule {            //定时任务
  Name =dbscd         //这个应用到了 job 里面
  Run = Full 1st sun at 23:05
  Run = Differential 2nd-5th sun at 23:05
  Run = Incremental mon-sat at 23:05
}

Catalog {
  Name =dbcatalog     //这个应用到了 job 里面
  dbname = "bacula"; dbuser = "bacula"; dbpassword = ""
  DB Address=10.10.2.226    //不要用 localhost,数据库所在的 IP
  DB Port=3306
}

Messages {
  Name = Daemon       //这个应用到了 job 里面
  mailcommand = "/sbin/bsmtp -h localhost -f \"\(Bacula\) \<%r\>\" -s \"Bacula daemon
message\" %r"
  mail = root@localhost = all, !skipped
  console = all, !skipped, !saved
  append = "/var/bacula/working/log" = all, !skipped
```

```
}

Console {
  Name = linux-0fdr-mon
  Password = "MJ8SyFywMm+4ntJf2EupggRWIWE6LozmLfIKC8J9nYt7"
  CommandACL = status, .status
}

##################################################################
#########
```

SD 配置:

```
Storage {
  Name =dbsd        //这里要与 director 配置文件的 storage 的 name 一样
  SDAddress=10.10.2.226   //这里是 SD 的 IP
  SDPort = 9103
  WorkingDirectory = "/var/bacula/working"
  Pid Directory = "/var/run"
  Maximum Concurrent Jobs = 20   //job 的最大连接数
}
Director {
  Name =linux-0fdr-dir     //这里要与 directory 的 name 相同
  Password = "5jipKA7pekDel6BUjglwD3LkPPBgxcecbLpLfl9puj0c"  //这里与 directory 中
storage 的 password 相同
}
Director {            //这个是做监控的配置
  Name = linux-0fdr-dir
  Password = "MJ8SyFywMm+4ntJf2EupggRWIWE6LozmLfIKC8J9nYt7"
  Monitor = yes

Device {
  Name =dbdev        //这里要与 directory 中 storage 的 device 一样
```

```
  Media Type = File   //这里要与 diretory 中 storage 的 Media Type 相同
  Archive Device =/home/bakcup/      //备份的路径
  LabelMedia = yes;
  Random Access = Yes;
  AutomaticMount = yes;
  RemovableMedia = no;
  AlwaysOpen = no;
}
Messages {
  Name = Standard
  director = linux-0fdr-dir = all
}
```

console 端的配置:

```
Director {
  Name = linux-0fdr-dir     //这里要与 directory 的 name 相 同
  DIRport = 9101
  address = 10.10.2.226     //console 端的 IP
  Password = "MJ8SyFywMm+4ntJf2EupggRWIWE6LozmLfIKC8J9nYt7"
          //这个密码要与 directory 的 password 一样
}
```

4.启动 bacula 的进程

在 server 端,只启动 directory 和 storage 的进程,启动命令如下:

```
#./bacula-ctl-dir  start
#./bacula-ctl-sd  start
```

启动之后,要给 bacula 添加存储介质,

```
linux-0fdr:/etc/bacula # ./bconsole
Connecting to Director 10.10.2.226:9101
1000 OK: linux-0fdr-dir Version: 2.4.3 (10 October 2008)
Enter a period to cancel a command.
```

*label

Automatically selected Catalog: dbcatalog

Using Catalog "dbcatalog"

Automatically selected Storage: dbsd

Enter new Volume name: **woyun** //这里随便给他起个卷名

Automatically selected Pool: dbpool

Connecting to Storage daemon dbsd at 10.10.2.226:9103 ...

Sending label command for Volume "test" Slot 0 ...

3000 OK label. VolBytes=188 DVD=0 Volume="test" Device="dbdev" (/home/bakcup/)

Catalog record for Volume "test", Slot 0 successfully created.

Requesting to mount dbdev ...

3906 File device "dbdev" (/home/bakcup/) is always mounted.

You have messages.

注意:上面黄色底红色字的为要输入的内容.

二、 bacula 在 client 服务器上的安装和配置

1.安装 bacula

#tar zxvf bacula-2.4.3.tar.gz

cd bacula-2.4.3

#./configure --enable-client-only

#make&&make install

默认安装在/etc/bacula/目录下

2.bacula 的配置

#more bacula-fd.conf

##

FD 配置:

Director

Name = linux-0fdr-dir //这里要与 directory 的 name 相同

Password = "Nd+Yuqe+Kd5wBc1S9uMTEGTNqNIe1cupnfuOdwboej6Z"//这里与 directory 中 client 的 password 相同

```
}

Director {            //这个是做监控的配置
  Name = linux-0fdr-mon
  Password = "MJ8SyFywMm+4ntJf2EupggRWIWE6LozmLfIKC8J9nYt7"
  Monitor = yes
}

FileDaemon {
  Name = dbfd          //这里要与 diretory 的 job 里的 client 相同
  FDport = 9102
  FDAddress=10.10.2.162   //被备份服务器的 IP
  WorkingDirectory = /var/bacula/working
  Pid Directory = /var/run
  Maximum Concurrent Jobs = 20
}
```

3.启动 bacula 的 fd 进程

```
  #bacula-ctl-fd start
```

三.备份以及还原过程

经过艰辛万苦终于走到了这一步,不容易啊,好现在备份下 10.10.2.162 的/home/bacula 这个目录,操作如下:

```
#./bacula

Connecting to Director 10.10.2.226:9101
1000 OK: linux-0fdr-dir Version: 2.4.3 (10 October 2008)
Enter a period to cancel a command.
*run
Automatically selected Catalog: dbcatalog
Using Catalog "dbcatalog"
A job name must be specified.
```

The defined Job resources are:

1: dbjob

2: restrory

Select Job resource (1-2): 1

Run Backup job

JobName: dbjob

Level: Full

Client: dbfd

FileSet: dbfs

Pool: dbpool (From Job resource)

Storage: dbsd (From Job resource)

When: 2009-08-25 15:23:14

Priority: 10

OK to run? (yes/mod/no): yes

Job queued. JobId=54

*list jobs

54 | dbjob | 2009-08-25 15:23:18 | B | F | 0 | 0 | R | //这里会出现一个列表,我当前只复制

最新的一个 JOBID,这里注意红色的 R 表示已经在运行了,要是 E 或者 C 就是有问题了，要是 T 就表示复制完成了.

*status

Status available for:

1: Director

2: Storage

3: Client

4: All

Select daemon type for status (1-4): 3

Automatically selected Client: dbfd

Connecting to Client dbfd at 10.10.2.162:9102

…… //这里省略一部分输出信息

54 Full 1,628 9.773 M OK 25- 8?009 15:27 dbjob

//看到这里已经显示 OK 了,表示备份成功

现在去看看备份路径下面是否生成了备份文件

du -sh *

9.7M　woyun

可以看到这个卷有 9.4M 的大小了,原来是 4K 的文件,再次证明了备份成功了.

下面做还原的操作,我们把 client 端的被备份目录下面文件进行删除

[root@localhost bacula]# rm -rf *

[root@localhost bacula]# ls

然后我们到 directory 端进行还原

./bconsole

Connecting to Director 10.10.2.226:9101

1000 OK: linux-0fdr-dir Version: 2.4.3 (10 October 2008)

Enter a period to cancel a command.

*restore

Automatically selected Catalog: dbcatalog

Using Catalog "dbcatalog"

To select the JobIds, you have the following choices:

1: List last 20 Jobs run

2: List Jobs where a given File is saved

3: Enter list of comma separated JobIds to select

4: Enter SQL list command

5: Select the most recent backup for a client

6: Select backup for a client before a specified time

7: Enter a list of files to restore

8: Enter a list of files to restore before a specified time

9: Find the JobIds of the most recent backup for a client

10: Find the JobIds for a backup for a client before a specified time

11: Enter a list of directories to restore for found JobIds

12: Cancel

Select item: (1-12): 5

Automatically selected Client: dbfd

Automatically selected FileSet: dbfs

```
+-------+-------+----------+-----------+---------------------+------------+
| JobId | Level | JobFiles | JobBytes  | StartTime           | VolumeName |
+-------+-------+----------+-----------+---------------------+------------+
|    54 | F     |    1,628 | 9,773,751 | 2009-08-25 15:23:18 | woyun      |
+-------+-------+----------+-----------+---------------------+------------+
You have selected the following JobId: 54
Building directory tree for JobId 54 ... ++++++++++++++++++++++++++++++++++++++++++++++++++++++
1 Job, 1,621 files inserted into the tree.
You are now entering file selection mode where you add (mark) and
remove (unmark) files to be restored. No files are initially added, unless
you used the "all" keyword on the command line.
Enter "done" to leave this mode.
cwd is: /
$ mark home
1,628 files marked.
$ done
Bootstrap records written to /var/bacula/working/linux-0fdr-dir.restore.1.bsr
The job will require the following
   Volume(s)                 Storage(s)                SD Device(s)
===========================================================================
   woyun                     dbsd                      dbdev
1,628 files selected to be restored.
Run Restore job
JobName:         restrory
Bootstrap:       /var/bacula/working/linux-0fdr-dir.restore.1.bsr
Where:           /home/bacula/
Replace:         always
FileSet:         dbfs
Backup Client:   dbfd
Restore Client:  dbfd
```

Storage: dbsd

When: 2009-08-25 16:09:03

Catalog: dbcatalog

Priority: 10

OK to run? (yes/mod/no): **mod**

Parameters to modify:

1: Level

2: Storage

3: Job

4: FileSet

5: Restore Client

6: When

7: Priority

8: Bootstrap

9: Where

10: File Relocation

11: Replace

12: JobId

Select parameter to modify (1-12): **9**

Please enter path prefix for restore (/ for none): /home/bacula

Run Restore job

JobName: restrory

Bootstrap: /var/bacula/working/linux-0fdr-dir.restore.1.bsr

Where: /home/bacula

Replace: always

FileSet: dbfs

Backup Client: dbfd

Restore Client: dbfd

Storage: dbsd

When: 2009-08-25 16:09:03

Catalog: dbcatalog

Priority: 10

OK to run? (yes/mod/no): yes

Job queued. JobId=55

*exit

OK 还原完成,我们现在到 client 端看看有什么还原回来

[root@localhost bacula]# ls

bacula-2.4.3 bacula-2.4.3.tar.gz

看到了已经还原回来了,说明我们的还原操作已经实现了。

为什么选择 Scala

ChinaUnix 网友：VWPOLo

如果你是一名 Java 程序员，并且关注这编程语言方面的发展，比如经常去 TIOBE 网站了解编程语言流行度排行，那么你应该听说过 Scala，如果你还没有开始学习 Scala，或者打算下个礼拜开始学的话，请先看看下面这篇文章，看看能不能改变你的想法。下面的内容为 Programming In Scala 这本书的节选，到目前为止，国内好像还没引进，你可以从亚马逊上面购买有国内的朋友翻译了其中的前 11 章，真是非常感谢)，

Scala 是为你准备的吗？你必须自己看明白并做决定。除了伸展性之外，我们发现喜欢用 Scala 编程实际上还有很多理由。最重要的四个将在本节讨论的方面该是：兼容性，简短，高层级抽象和高级的静态类别。

Scala 是兼容的

Scala 不需要你从 Java 平台后退两步然后跳到 Java 语言前面去。它允许你在现存代码中加点儿东西——在你已有的东西上建设——因为它被设计成无缝地与 Java 实施互操作。Scala 程序会被编译为 JVM 的字节码。它们的执行期性能通常与 Java 程序一致。Scala 代码可以调用 Java 方法，访问 Java 字段，继承自 Java 类和实现 Java 接口。这些都不需要特别的语法，显式接口描述，或粘接代码。实际上，几乎所有 Scala 代码都极度依赖于 Java 库，而经常无须在程序员意识到这点。

交互式操作的另一个方面是 Scala 极度重用了 Java 类型。Scala 的 Int 类型代表了 Java 的原始整数类型 int，Float 代表了 float，Boolean 代表 boolean，等等。Scala 的数组被映射到 Java 数组。Scala 同样重用了许多标准 Java 库类型。例如，Scala 里的字串文本"abc"是 java.lang.String，而抛出的异常必须是 java.lang.Throwable 的子类。

Scala 不仅重用了 Java 的类型，还把它们“打扮”得更漂亮。例如，Scala 的字串支持类似于 toInt 和 toFloat 的方法，可以把字串转换成整数或者浮点数。因此你可以写 str.toInt 替代 Integer.parseInt(str)。如何在不打破互操作性的基础上做到这点呢？Java 的 String 类当然不会有 toInt 方法。实际上，Scala 有一个解决这种高级库设计和互操作性不相和谐的通用方案。Scala 可以让你定义**隐式转换：implicit conversion**，这常常用在类型失配，或者选用不存在的方法时。在上面的例子里，当在字串中寻找 toInt 方法时，Scala 编译器会发现 String 类里没有这种方法，但它会发现一个把 Java 的 String 转换为 Scala 的 RichString 类的一个实例的隐式转换，里面定义了这么个方法。于是在执行 toInt 操作之前，转换被隐式应用。

Scala 代码同样可以由 Java 代码调用。有时这种情况要更加微妙，因为 Scala 是一种比 Java 更丰富的语言，有些 Scala 更先进的特性在它们能映射到 Java 前需要先被编码一下。第 29 章说明了其中的细节。

Scala 是简洁的

Scala 程序一般都很短。Scala 程序员曾报告说与 Java 比起来代码行数可以减少到 1/10。这有可能是个极限的例子。较保守的估计大概标准的 Scala 程序应该有 Java 写的同样的程序一半行数左右。更少的行数不仅意味着更少的打字工作，同样意味着更少的话在阅读和理解程序上的努力及更少的出错可能。许多因素在减少代码行上起了作用。

首先，Scala 的语法避免了一些束缚 Java 程序的固定写法。例如，Scala 里的分号是可选的，且通常不写。Scala 语法里还有很多其他的地方省略了东西。比方说，比较一下你在 Java 和 Scala 里是如何写类及构造函数的。在 Java 里，带有构造函数的类经常看上去是这个样子：

```
// 在 Java 里
class MyClass {
  private int index;
  private String name;
  public MyClass(int index, String name) {
    this.index = index;
    this.name = name;
  }
}
```

在 Scala 里，你会写成这样：

```
class MyClass(index: Int, name: String)
```

根据这段代码，Scala 编译器将制造有两个私有成员变量的类，一个名为 index 的 Int 类型和一个叫做 name 的 String 类型，还有一个用这些变量作为参数获得初始值的构造函数。这个构造函数还将用作为参数传入的值初始化这两个成员变量。一句话，你实际拿到了与罗嗦得多的 Java 版本同样的功能。Scala 类写起来更快，读起来更容易，最重要的是，比 Java 类更不容易犯错。

有助于 Scala 的简洁易懂的另一个因素是它的类型推断。重复的类型信息可以被忽略，因此程序变得更有条理和易读。

但或许减少代码最关键的是因为已经存在于你的库里而不需要写的代码。Scala 给 了你许多工具来定义强有力的库让你抓住并提炼出通用的行为。例如，库类的不同方面可以被分成若干特质，而这些有可以被灵活地混合在一起。或者，库方法可以 用操作符参数化，从而让你有效地定义那些你自己控制的构造。这些构造组合在一起，就能够让库的定义既是高层级的又能灵活运用。

Scala 是高层级的

程序员总是在和复杂性纠缠。为了高产出的编程，你必须明白你工作的代码。过度复杂的代码成了很多软件工程崩溃的原因。不幸的是，重要的软件往往有复杂的需求。这种复杂性不可避免；必须（由不受控）转为受控。

Scala 可以通过让你提升你设计和使用的接口的抽象级别来帮助你管理复杂性。例如，假设你有一个 String 变量 name，你想弄清楚是否 String 包含一个大写字符。在 Java 里，你或许这么写：

```
// 在 Java 里
boolean nameHasUpperCase = false;
for (int i = 0; i < name.length(); ++i) {
  if (Character.isUpperCase(name.charAt(i))) {
    nameHasUpperCase = true;
    break;
  }
}
```

在 Scala 里，你可以写成：

```
val nameHasUpperCase = name.exists(_.isUpperCase)
```

Java 代码把字串看作循环中逐字符步进的低层级实体。Scala 代码把同样的字串当作能用**论断：predicate** 查询的字符高层级序列。明显 Scala 代码更短并且——对训练有素的眼睛来说——比 Java 代码更容易懂。因此 Scala 代码在通盘复杂度预算上能极度地变轻。它也更少给你机会犯错。

论断，_.isUpperCase，是一个 Scala 里面函数式文本的例子。它描述了带一个字符参量（用下划线字符代表）的函数，并测试其是否为大写字母。

原则上，这种控制的抽象在 Java 中也是可能的。为此需要定义一个包含抽象功能的方法的接口。例如，如果你想支持对字串的查询，就应引入一个只有一个方法 hasProperty 的接口 CharacterProperty：

```
// 在 Java 里
interface CharacterProperty {
  boolean hasProperty(char ch);
}
```

然后你可以在 Java 里用这个接口格式一个方法 exists：它带一个字串和一个 CharacterProperty 并返回真如果字串中有某个字符符合属性。然后你可以这样调用 exists：

```
// 在 Java 里
exists(name, new CharacterProperty {
  boolean hasProperty(char ch) {
    return Character.isUpperCase(ch);
  }
});
```

然而，所有这些真的感觉很重。重到实际上多数 Java 程序员都不会惹这个麻烦。他们会宁愿写个循环并漠视他们代码里复杂性的累加。另一方面，Scala 里的函数式文本真地很轻量，于是就频繁被使用。随着对 Scala 的逐步了解，你会发现越来越多定义和使用你自己的控制抽象的机会。你将发现这能帮助避免代码重复并因此保持你的程序简短和清晰。

Scala 是静态类型的

静态类型系统认定变量和表达式与它们持有和计算的值的种类有关。Scala 坚持作为一种具有非常先进的静态类型系统的语言。从 Java 那样的内嵌类型系统起步，能够让你使用**泛型：generics** 参数化类型，用**交集：intersection** 联合类型和用**抽象类型：abstract type** 隐藏类型的细节。这些为建造和组织你自己的类型打下了坚实的基础，从而能够设计出即安全又能灵活使用的

接口。

如果你喜欢动态语言如 Perl，Python，Ruby 或 Groovy，你或许发现 Scala 把 它的静态类型系统列为其优点之一有些奇怪。毕竟，没有静态类型系统已被引为动态语言的某些主要长处。绝大多数普遍的针对静态类型的论断都认为它们使得程序 过度冗长，阻止程序员用他们希望的方式表达自己，并使软件系统动态改变的某些模式成为不可能。然而，这些论断经常针对的不是静态类型的思想，而是指责特定 的那些被意识到太冗长或太不灵活的类型系统。例如，Alan Kay，Smalltalk 语言的发明者，有一次评论：“我不是针对类型，而是不知道有哪个没有完痛的类型系统，所以我还是喜欢动态类型。”

我们希望能在书里说服你，Scala 的 类型系统是远谈不上会变成“完痛”。实际上，它漂亮地说明了两个关于静态类型通常考虑的事情（的解决方案）：通过类型推断避免了赘言和通过模式匹配及一些 新的编写和组织类型的办法获得了灵活性。把这些绊脚石搬掉后，静态类型系统的经典优越性将更被赏识。其中最重要的包括程序抽象的可检验属性，安全的重构， 以及更好的文档。

可检验属性：静态类型系统可以保证消除某些运行时的错误。例如，可以保证这样的属性：布尔型不会与整数型相加；私有变量不会从类的外部被访问；函数带了正确个数的参数；只有字串可以被加到字串集之中。

不过当前的静态类型系统还不能查到其他类型的错误。比方说，通常查不到无法终结的函数，数组越界，或除零错误。同样也查不到你的程序不符合式样书（假设有 这么一份式样书）。静态类型系统因此被认为不很有用而被忽视。舆论认为既然这种类型系统只能发现简单错误，而单元测试能提供更广泛的覆盖，又为何自寻烦恼 呢？我们认为这种论调不对头。尽管静态类型系统确实不能替代单元测试，但是却能减少用来照顾那些确需测试的属性的单元测试的数量。同样，单元测试也不能替代静态类型。总而言之，如 Edsger Dijkstra 所说，测试只能证明存在错误，而非不存在。因此，静态类型能给的保证或许很简单，但它们是无论多少测试都不能给的真正的保证。

安全的重构：静 态类型系统提供了让你具有高度信心改动代码基础的安全网。试想一个对方法加入额外的参数的重构实例。在静态类型语言中，你可以完成修改，重编译你的系统并 容易修改所有引起类型错误的代码行。一旦你完成了这些，你确信已经发现了所有需要修改的地方。对其他的简单重构，如改变方法名或把方法从一个类移到另一 个，这种确信都有效。所有例子中静态类型检查会提供足够的确认，表明新系统和旧系统可以一样的工作。

文档：静态类型是被编译器检查过正确性的程序文档。不像普通的注释，类型标注永远都不会过期（至少如果包含它的源文件近期刚刚通过编译就不会）。更进一步说，编译 器和集成开发环境可以利用类型标注提供更好的上下文帮助。举例来说，集成开发环境可以通过判定选中表达式的静态类型，找到类型的所有成员，并全部显示出 来。

虽然静态类型对程序文档来说通常很有用，当它们弄乱程序时，也会显得很讨厌。标准意义上来说，有用的文档是那些程序的读者不可能很容易地从程序中自己想出来的。在如下的方法定义中：

```
def f(x: String) = ...
```

知道 f 的变量应该是 String 是有用的。另一方面，以下例子中两个标注至少有一个是讨厌的：

```
val x: HashMap[Int, String] = new HashMap[Int, String]()
```

很明显，x 是以 Int 为键，String 为值的 HashMap 这句话说一遍就够了；没必要同样的句子重复两遍。

Scala 有非常精于此道的类型推断系统，能让你省略几乎所有的通常被认为是讨厌的类型信息。在上例中，以下两个不太讨厌的替代品也能一样工作：

val x = new HashMap[Int, String]()

val x: Map[Int, String] = new HashMap()

Scala 里的类型推断可以走的很远。实际上，就算用户代码丝毫没有显式类型也不稀奇。因此，Scala 编 程经常看上去有点像是动态类型脚本语言写出来的程序。尤其显著表现在作为粘接已写完的库控件的客户应用代码上。而对库控件来说不是这么回事，因为它们常常 用到相当精妙的类型去使其适于灵活使用的模式。这很自然。综上，构成可重用控件接口的成员的类型符号应该是显式给出的，因为它们构成了控件和它的使用者间 契约的重要部分。

IP 校验和详解

ChinaUnix 网友：bripengandre

一、校验和算法

之前一直只知道 IP 校验和算法反码求和相关的,但具体细节不清楚,今天了解了下。

IP 校验和主要是用来保证数据(IP 包头)的完整性的.它用的算法非常简单,就是反码求和校验.需要注意的是反码求和又叫 1 的补码(one'scomplement),而 2 的补码就是我们通常说的补码求和了.校验算法具体如下。

1、发送方

i)将校验和字段置为 0,然后将 IP 包头按 16 比特分成多个单元,如包头长度不是 16 比特的倍数,则用 0 比特填充到 16 比特的倍数;

ii)对各个单元采用反码加法运算(即高位溢出位会加到低位,通常的补码运算是直接丢掉溢出的高位),将得到的和的反码填入校验和字段;

iii)发送数据包.

2、接收方

i)将 IP 包头按 16 比特分成多个单元,如包头长度不是 16 比特的倍数,则用 0 比特填充到 16 比特的倍数;

ii)对各个单元采用反码加法运算,检查得到的和是否符合是全 1(有的实现可能对得到的和会取反码,然后判断最终值是不是全 0);

iii)如果是全 1 则进行下步处理,否则意味着包已变化从而丢弃之.

需要强调的是反码和是采用高位溢出加到低位的,如 3 比特的反码和运算:100b+101b=010b(因为 100b+101b=1001b,高位溢出 1,其应该加到低位,即 001b+1b(高位溢出位)=010b),具体细节请参考文章:http://blog.chinaunix.net/u/20/showart_438418.html

二、校验和源码

网上流传多组实现,常见的有如下两种(如追求效率可改写为汇编代码):

1、RFC1071 源码

```
unsigned short csum(unsigned char *addr, int count)
{
    /* Compute Internet Checksum for "count" bytes
     * beginning at location "addr".
    */
    register long sum = 0;

    while( count > 1 )

    {
       /* This is the inner loop */
       sum += * (unsigned short) addr++;
       count -= 2;
     }

    /* Add left-over byte, if any */
    if( count > 0 )
       sum += * (unsigned char *) addr;

    /* Fold 32-bit sum to 16 bits */
    while (sum>>16)
       sum = (sum & 0xffff) + (sum >> 16);

    return ~sum;
}
```

第一个 while 循环是做普通加法(2 进制补码加法),因为 IP 包头和 TCP 整个报文段比较短(没达到 2^17 数量级),所以不可能导致 4 字节的 sum 溢出(unsigned long 一般至少为 4 字节)).

紧接着的一个判断语句是为了能处理输入数据是奇数个字节的这种情况.

再接着的数据循环是实现反码算法(在前面的普通加法得到的数据的基础上),由反码和的高位溢出加到低位的性质,可得到"32 位的数据的高位比特移位 16 比特,再加上原来的低 16 比特,不影响最终结果" 这个等价运算,因为 sum 的最初值(刚开始循环时)可能很大,所以这个等价运算需循环进行,直到 sum 的高比特(16 比特以上)全为 0.对于 32 位的 sum,事实上这个运算循环至多只有两轮,所以也有程序直接用两条"sum = (sum & 0xffff) + (sum >> 16);"代替了整个循环.

最后,对和取反返回.

2、对数据长度没限制的实现

```
unsigned short cksum (struct ip *ip, int len)
{
    long sum = 0; /* assume 32 bit long, 16 bit short */
```

```
    while ( len >1 )
    {
      sum += *((unsigned short *) ip)++;
      if (sum & 8x00000000) /* if high-order bit set, fold */
         sum = (sum & 0xFFFF) + (sum>> 16) ;
      len -= 2;
    }
    if ( len ) /* take care of left over byte */
       sum += ( unsigned short ) * (unsignedl char *) ip;
    while ( sum >> 16)
      sum =(sum & 0xFFFF) + (sum>> 16);
    return ~sum;
}
```

这个实现与前面的一个的最大的不同是对数据的长度没什么限制了,因为它在第一个循环的加法运算中实时检测 sum 的高位的值,一旦发现其有溢出的危险,就及时运用等价运算关系消除了这个危险.

三、几个细节问题

1、数据部分改变时的重校验

考虑这样的应用场景:路由器转发 IP 报文时,有可能只更改了 IP 数据包头的部分内容(如更改了 TTL,分片了或 SNAT 更改了源 IP 等~~~),却需要重校验的问题.为提高转发效率,要求重校验算法尽可能快,故出现了如下所示的重校验算法（只是一个简单的示例）：

```
UpdateTTL(struct ip_hdr *ipptr, unsigned char n)
{
   unsigned long sum;
   unsigned short old;

   old = ntohs(*(unsigned short *)&ipptr->ttl);
   ipptr->ttl -= n;
   sum = old + (~ntohs(*(unsigned short *)&ipptr->ttl) & 0xffff);
   sum += ntohs(ipptr->Checksum);
   sum = (sum & 0xffff) + (sum>>16);
   ipptr->Checksum = htons(sum + (sum>>16));
```

```
}
```

算法的实现依据是这样的.假设包头原校验和为~C,改变的字段的原始值是 m,更改后的值是 m',设~C'为重校验和,则有 ~C' = ~(C+(-m)+m') = ~C+(m-m') = ~C+m+~m'

等价关系的成立基于反码的运算性质:取反运算满足结合律,按位取反运算与符号取反(及相反数)是等价的(即~C=-C).

如果有多个字段改变，只是上面的公式中的 m 和 m'有多个而已，直接用反码加法搞定即可。

2、为什么采用反码和运算

IP 数据包校验要求速度快,所以只采用了简单的和校验,为什么采用反码和而不是补码和呢?

i)反码和的溢出有后效性(蔓延性)

反码和将高位溢出加到低位,导致这个溢出会对后面操作有永久影响,有后效性;而补码和直接将高位和溢出,导致这个溢出对后面的操作再无影响,因此无后效性

ii)反码校验无需考虑字节序

正因为反码和的溢出有后效性,导致大端字节序(big-endian)和小端字节序(little-endian)对同一数据序列(如两个 16 比特的序列)产生的校验和也只是字节序相反，而补码和因为将溢出丢掉了,不同字节序之间的校验和大不相同且没什么联系。

基于以上的理由，校验和运算既可选择在数据被转换成网络字节序前，也可选择在之后，只要保证被校验的字段和填写的校验和字段的字节序保持一致就可以了。（这其实可以看作是负负得正，计算校验和与字节序有关，然后写校验和字段与字节序有关，然后直接计算校验和再写校验和字段则与字节序无关。）

四、参考文章

http://blog.chinaunix.net/u/20/showart_438512.html，关于 IP 校验和的
http://blog.chinaunix.net/u/12313/showart_176114.html，关于网络校验和的
http://www.wesoho.com/article/Delphi/2143.htm，关于 IP 校验和的
http://blog.chinaunix.net/u/20/showart_438418.html，关于补码和反码的

网友热评

****

热点技术评论

一道 C 语言指针运算的基础题

char *(*(* var)())[10]表示什么

scanf 输入字符串如何“按需分配”空间

grep 正则表达式{n,m}的问题

800MB/s 的数据量，请大家帮忙看看

看看大家对数组和指针的了解

vim 的问题

用 OpenBSD 做的路由防火墙

MySQL 和 PostgreSQL 在 FreeBSD 中的优劣

一个大的结构体如何写入文本文件?

嵌入式 web server 是做什么用的?

大家测过 64linux vs 32windows 的性能没有

a=a+b 与 a+=b 差别在哪?

有人对 SMTP 协议很熟悉的吗?

unix 大文件处理编程，估计要 10G 的日志

apache 安装问题

请问 init 3 命令是什么意思

内核恐慌了，这是咋回事?

tcpdump 无数据显示

各位重启系统用什么命令?

安装 redhat5 出现的问题

RedHat cacti 安装问题求救!

请问一下 TCP 包的重组过程

HTTP server 对于重传的 GET 包如何处理?

分享一个我写的最简单 ARM-Linux Bootloader

热点新闻评论

劣币是如何驱逐良币的

桌面无用? 界面不重要?

别碰我，VIM(转)

关于 30 岁学习 c 和 c++

谁不会成为 Linux（ubuntu）的用户

从职场的阴暗面看我们的未来

Java 面临终结 Scala 将取而代之

Windows 7，一个生不逢时的悲剧

DDos 攻击或与俄罗斯格鲁吉亚冲突有关

一名合格的存储工程师要具备那些技能啊?

Ylmf Linux Y1.15(Ubuntu)雨林木风开源新作

算法都没定型，其他设计全都是空谈

实践再次证明光会 C 不会 C++找工作很难

Infoworld 评出有史以来 10 大最具价值开源软件

DBA 之路该继续吗?

大学生学好高等数学很重要

黑客再爆 Linux 内核高危漏洞

现在 linux 系统哪个版本的性能更强大一些?

袁萌：Linux 挽救了 Windows 一条命

使用 linux 二年后却陷入了新手般的问题

Linux 没有注册表?

一个命令就可以精通 linux 系统

CentOS 算不算是一个盗版操作系统?

Linux 源代码已超过 1150 万行 红帽贡献最大

C 语言里为什么有这么多没用的特性?